QL TECHNICAL REVIEW

ISSUE 8 PRICE £1.50

CONTENTS

PUBLISHED BY C.G.H. SERVICES,
CWM GWEN HALL, PENCADER, DYFED,
CYMRU, SA39 9HA

EDITORIAL / NEWS

It has taken longer than I had anticipated to compile this issue of QLTR, sorry for the delay. I hope the wait is worth it.

Congratulations to Dilwyn (Dilwyn Jones Computing) and Janet on the birth of their son Gareth Sion. I wish them all the best for the future.

If you fancy doing a review/article you can contact me at 57 Shaftesbury Road, Romford, Essex, RM1 2QJ or on Tony Firshman's Bulletin Board. Submissions for the magazine are better on disk/microdrive in Quill Doc format with an accompanying printout of the article. All disks/microdrives will be returned.

Finally many thanks to Miracle systems for repairing my twin disk drives promptly and not charging even though the drives must be five years old. Do any other firms provide a lifetime guarantee of their products ?

INTERNATIONAL QL MEETING

The Dutch QL-User club SIN_QL_AIR and the German QL-User club SQLUC have organised an International QL Meeting to be held in Eindhoven/Netherlands on February 27th 1993 between 10 am and 5 pm. This meeting is entirely dedicated to the QL and QDOS (but of course Minerva and SMS2 are also welcome!). The venue is Eindhoven, Southern Holland, Roostenlaan 296, St Joris college. Admission is FL 3,50 pp (includng a free drink!). Table's for commercial use are FL 15 (2M long). For more information you can contact the organiser: c/o J.J. v/d Molengraaf, Mullerweg 17 5624 JC, Eindhoven, Holland, TEL: 31-40-442309 or Franz Herrmann, Talstrasse 21 W-5460 Ockenfels,Germany, TEL: 49-(0)2644-1855 (only on weekends) or Marco Holmer, J.P. Coenstraat 61 bis,3531 EN Utrecht, Holand, TEL: 31-30-948673.

HARDWARE

Miracle Systems are still working on their graphics board (actually it will replace the whole QL with the Goldcard attached), PC/QL board and SCSI Interface. There maybe more news early in 1993. In the meantime they have re-launched their centronics interface with all the hardware enclosed in the centronics hood, price is £25.

EEC have now modified the MGT Lifetime drives, in particular, changing the regulator from 1 to 2 amps, so there should be no more incompatibility problems. They are also selling 2Mb drives for £90 each or £170 for 2 units. Jürgen Falkenberg products are now being distributed by EEC and include a 32M Hard disk drive for £275, write for further details/products. They are also supplying a serial mouse which requires no QIMI interface as it is driven by some special software sold by Jochen Merz Software, price is £45.

CL systems who produce the CQV1 real time video digitizer have now made it Gold Card Compatible, an upgrade is available for £30 to UK users.

Hermes is a new replacement for the QL co-processor (IC 8049 IPC) available from TF Services. It provides reliable serial output up to 19200 bps, independent baud rates for the serial ports, stops keyboard 'bounce', improves 'fuzzy' and 'random' sound , gives a key click , provides extra input and output lines and the Reset/INT7 is invoked more safely. Price is £25.

Trump card production has now been taken over by Qubbesoft. The new price is £95.00 + P&P for a 768K Trump card. Between the 1st of January and the 28th of February there will be a special offer on bulk purchases by any user group. You will be able to buy a minimum of 5 Trump cards for £90 each + P+P or between six and ten Trump cards for £85 each +P&P. Qubbesoft also produce 512K and 0K ExpandeRAM's priced at £45 and £20 + P&P respectively and twin 720k disk drives for £100 + P&P. A product which has not been available for a while but is now being re-launched is the QEP3 Eprom programmer. This can program 2716 to 275124 64K eproms. The price is £120 + £3 P&P which includes an A5 manual and a years guarantee.

QVME - the QL-Emulator for the Mega STE has now arrived via Jochen Merz Software. No soldering is needed and you can use screen resolutions up to 1024 X 780. It has a host of other features - write for more details, the price is £256 +£4.50 P&P for Europe (the price excludes VAT E&OE).

SOFTWARE

PROGS have written in to reply to the comments on the review of DATAdesign in QLTR7 saying that most of the problems/missing features have been corrected in the latest release of version 2. If you are a user of DATAdesign and have any ideas for improvements etc. please write a letter to PROGS as they say in their letter "Datadesign is still under development and we try to help everybody ".

DI-REN hve been busy releasing the PC version of FLEET TACTICAL COMMAND II which can be networked to a QL via a simple serial port link (cable available from TF Services). It features improved graphics and having seen it recently at a QUANTA workshop I certainly recommend it to anyone with access to a PC. The price is £69.95. They have also released a QL - PC Fileserver which allows you to link a QL to a PC and then access the PC files in the same manner as QL storage devices. All standard QL file operations are supported, the price is £24.50. These are available through DJC. We will have a review of this in the next issue.

C.G.H. SERVICES has now released QUICK MANDELBROT III ,price £15, and FRACTALS FROM NEWTON'S POLYNOMIAL METHOD priced at £12.50. SToQL the ST screen transfer and image processing program has now been made GOLD card compatible.

JOCHEN MERZ SOFTWARE is now available in the UK through Pointer Products. New releases are QSpread, a spreadsheet for the Pointer Environment allowing more than 32000 cells, SER Mouse V2, a software driver which lets you connect a serial mouse to one of the SER ports of the QL that mimics the QIMI interface and THE LONELY JOKER, a solitaire card game based on three different patients. Prices are £49 + P&P, £14.90 + £1.50 P&P and £14.90 + £1.50 P&P respectively.

DIGITAL PRECISION have produced special editions of PC CONQUEROR and LIGHTNING to take advantage of the memory and speed of the GOLD card together with releasing PERFECTION S/E. Prices are £99.95, £39.95 and £99.95 respectively.

Dilwyn Jones Computing are now selling the software formally sold by Care Electronics such as QPAC1 & QPAC2. New releases include the MEGA TOOLKIT, a comprehensive toolkit of over 200 BASIC extensions which can be used in your BASIC programs, BANTER, a banner making program which uses outline fonts for improved print quality and smooth edges to large letters and PRINTMASTER a printer control utility for sending all those easy to use and memorable ESC codes to your printer without having to reach for your manual. Prices are £25, £25 and £20 respectively.

All the best for the New Year,

Bruce Nicholls

MIRACLE SYSTEMS LTD
25 BROUGHTON WAY
OSBALDWICK
YORK
YO1 3BG
TEL: 0904 423986

(EEC) W.N.RICHARDSON & CO.
18-21 MISBOURNE HOUSE
CHILTERN HILL
CHALFONT ST. PETER
SL9 9UE
TEL: 0753 888866

CL SYSTEMS
403 CHAPTER ROAD
DOLLIS HILL
LONDON
NW2 5NG
TEL: 081 459 1351

TF SERVICES
12 BOUVERIE PLACE
LONDON
W2 1RB
TEL: 071 724 9053

QUBBESOFT P/D
38 BRUNWIN ROAD
RAYNE
BRAINTREE
ESSEX
CM7 5BU
TEL: 0376 347852

JOCHEN MERZ SOFTWARE
IM STILLEN WINKEL
12 - D-4100 DUISBURG 11
GERMANY
TEL: 0203 501274

PROGS
HAACHTSTRAAT 92
B-3020 VELTEM
BELGUIM
TEL: 016 48 89 52

DIGITAL PRECISION LTD
222 THE AVENUE
CHINGFORD
LONDON
E4 9SE
TEL: 081 527 5493

DILWYN JONES COMPUTING
41 BRO EMRYS
TAL-Y-BONT
BANGOR
GWYNEDD
LL57 3YT
TEL: 0248 354023

POINTER PRODUCTS
68 KINGDOWN ROAD
BLANDFORD
DORSET
DT11 8BN
TEL: 0258 455117

MEETING IN MÜNSTER

March the 21st was the date of the International QL Meeting, at Münster in Germany.

QL users from various countries all headed for Münster-Roxel, in North Western Germany. Franz Herrmann of the Sinclair QL User Club e.V. (of Germany) had made an excellent effort to contact visitors as early as the end of December, ensuring that overseas visitors had plenty of time to plan for this day.

Quanta had organised a mini-bus for this outing, and about a dozen of us set out (in my case, after a six hour car journey from Wales) from various pickup points starting from the Chairman's house in Raunds. Most of the Quanta officials were on board, and we were ably driven by the Chairman and General Secretary, who proved as adept at driving the mini-bus as driving Quanta. Some of the British traders had travelled by car on the ferry as well.

We sailed at 11pm from England and arrived about two and a half hours later in Dunkirk. We crossed to Belgium, then through The Netherlands and headed for Germany, grabbing what sleep we could on the way (fortunately, the Autobahn proved to be a gentler road than most British roads, although Ron did seem to manage to drive over the few pot-holes and drains we passed and jerk us back from our sleep).

We arrived at the venue shortly after the meeting opened at 8AM. Most of us had remembered to adjust our watches!

The venue was a large school hall. We strolled in half asleep and looked around before setting up. Quanta took a stand at the far corner of the hall (conveniently next to the coffee-bar!)

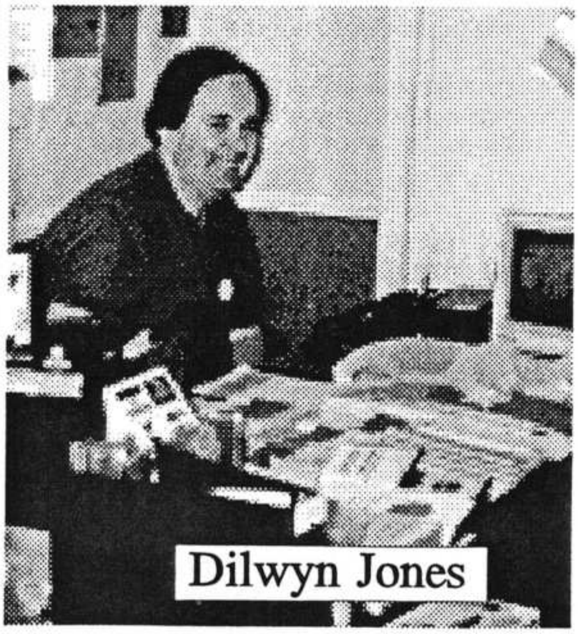
Traders present from Britain (most of whom had travelled separately) included Miracle Systems Ltd, TF Services, Qubbesoft, EEC Ltd and DJC. German traders included Jochen Merz, who displayed his large range of QL and ST-QL products (including many pointer environment products), Jurgen Falkenburg, who sells a large range of QL hardware, including a fairly low cost hard disk interface, QLympic Computer Systems (Martin Florichs) with a range of QL software, Wilfried Krummrey and Albin Hessler Software. Albin showed his EasyPtr II system (an aid for pointer environment program writers) and a serial mouse connected to the serial port of the QL, driven by special software to make it emulate a ptr mouse. David Johnson (on the EEC Ltd. stand) showed me an interesting little application he had written using Easy-Ptr for selecting control codes for a printer from menus instead of having to go to BASIC to send control codes for your printer after looking them up in a manual. It amply showed the advantage to programmers of using pointer environment and applications such as Easy-Ptr. For more details of Easy-Ptr, see QL Technical Review issue 7, page 17, for a review by Ian Bruntlett. He also reviews Jochen Merz's Menu Extensions in the same issue.

Joachim and Nathan of PROGS (Belgium) looked busy all day, demonstrating the latest version of DATA Design and The Painter (now with a HP Deskjet driver), while Urs König of Cowo Electronic, Switzerland, demonstrated his Q-Top system.

All that would have been impressive enough. But there were also the visitors and traders from other countries who made this a truly international meeting. There were members of Sin-QL-air (the Dutch QL users group), QItaly club, NASA (Norway), the Swedish QL group and I also met a few visitors from other countries too, although I cannot remember all the details!

Interestingly, even though it was a German venue, the language of the day seemed to be English. It enabled QL users from various countries to communicate quite easily (although I was more than once roped in to translate one English accent into another). I was quite amused to see that some German QL users label their disks in English, for example, and listening to conversations in German, the terminology largely seemed to be in English - it appears that the QL enables and promotes international communication quite well! Everyone was so friendly and relaxed - no international barriers here!

As a note in passing, CGH Services, who were not at the meeting, have an English/German dictionary as part of their PD library. That would have proved quite useful at times in a portable QL as we drove around the area looking for a fuel station, for example!



Dilwyn Jones

Several talks and demonstrations took place during the day on subjects as diverse as Fleet Tactical Command, Networking, Forth and BBS systems. I did not get to go to these, sadly, as I was manning a stand during most of the day (my wife had disappeared on a shopping trip with Kath Dunnett!). Thanks to Ralf Reköndt too, for his help with translation when required and for the loan of his ST-QL during the day.

Miracle Systems apparently had a very good day and sold goodness knows how many Gold Cards. The hardware traders seemed to be having a better day of it than the software traders, although there was plenty of interest in everything. Qubbesoft sold a range of public domain software and apparently received some new disks at the show, so we may be seeing yet more interesting disks of PD software soon from them.

There was an interesting item in the International QL Report magazine about ZX81 and Spectrum emulators from The Netherlands. These are apparently shareware programs, where you get a trial version then send a payment to the author for help and upgrades (registration). Maybe we will hear more of this at a later date. I have also heard that Lear Data Systems in Britain are developing a Spectrum emulator.

It was interesting to note the profusion of Atari ST-QL's at the meeting. They seem to be very popular in Germany, thanks to the work of people like Jochen Merz, who has done such a lot to promote the QL and compatibles in Europe. Ralf Reköndt's ST-QL is the only one I have ever used in anger (it was a pleasure really) and I was quite surprised by both the speed at which programs ran and at the compatibility level - during 8 hours of use at the show, not one program fell over! Also, it was quite a shock to be working in a 4MB environment. By the end of the day, there were so many programs present it became quite a task to CTRL-C until you found the program you wanted! I could get quite addicted to an ST-QL if I had one of my own. It might be a waste for me, though, since I would probably never use it in ST mode. The QL is my favourite computer (or should I say the QL is my favourite add on for a Gold Card?) and I have no plans to move to another computer at the moment. Ralf had brought his ST-QL over to Britain last year, but I did not get much time to play with it then. I know several users in Britain who have one (including 2 Quanta committee members) and seem quite pleased with it.

Although an emulator is available for the Amiga computer, to enable it to run most QL software, this did not seem as popular in Germany and I did not see any in use at the meeting.

I picked up no news from Miracle Systems Ltd on any new products they have planned for this year, although I didn't exactly press hard. We are all quite aware I am sure of rumours of an extended display board (see International QL Report magazine March/April 1992 issue) and even stories about a new QL (see article in QL World April 1992). Judging by the presence and activity at the meeting, the QL market is not only not declining, but seems to be actually expanding somewhat at the moment!

One thing which stuck out a mile (oops, sorry, kilometre) at this meeting was that the average German user is more likely to be using Pointer Environment than the average British user. Just about every ST-QL and QL at the meeting had pointer environment present, and I think we in Britain can learn a bit from this. Pointer Environment does take a little bit of time to get used to, but once you are familiar with it, struggling with a QL without it is a bit of a chore and the QL seems naked without it after a while. How I wish it had been on the QL from day 1! Software houses such as PROGS and Jochen Merz already produce programs using Pointer Environment (e.g. DATA Design). The one thing pointer environment needs is a better tutorial so that new users can learn to use it more easily. Although I am a great fan of Tony Tebby/QJump software, I do think his manuals are aimed at the more experienced and technically minded user and could do with more information for the less experienced user.

The day finished for use at 5pm (although the meeting was to continue until late at night) when we retired to the Hotel-Restaurant Brintrup to catch up on our sleep after an excellent meal. We set out the following morning for a long return journey to England and a well deserved rest after our long trip back. A slight blemish on the two days in Germany was the weather (it seemed as if we had taken the British wet weather with us to Germany!), but we did our best to ignore it!

I would like to thank Phil Borman of Quanta for both arranging the British trip and both to him and Ron Dunnett for all the driving they did during the weekend. We all had a lovely weekend and look forward to the next QL excuse for a trip to the continent.

Dilwyn Jones

FTIDY

A Simple File Handler compiled(Supercharge) for the Q.L.

This program is presented either on cartridge or disk and requires only the minimum memory of 128K. A single page passes as a manual of instruction such is the simplicity. As in all things simple, one must pay rather more attention to the learning effort to discover all that this program has to offer. For instance, there are two boot programs; one is to provide a 'backup' program which, when completed, is dropped off the 'backed-up' medium, leaving only the one 'boot' which provides all that is necessary to run the program. A menu is presented offering a choice one of which is a program 'APM Sweeper' having absolutely nothing to do with FTidy: it turns out to be a game. Splendid confusion!

Having selected the first item on the menu, FTidy128, a screenful of information is given by way of four screen-wide windows, two of which have borders which cleverly seperate them, the top being split into three sections giving details of the medium being used, its name, total and free memory in sectors on the left, the centre the name of the drive being used and the right-hand side giving the date and time and the number of files on the medium, up to 160. The time does not show seconds so it might be some time before it is realised that the clock is working. The main window gives a list of eight commands and what they do, together with date reset and a note to press 'h' for the help screen which turns out to be the window currently shown. This window is not active, except for the date/time reset 'd'. The lower window shows the default drives of source and target and between this one and the previous window is a borderless window containing the eight commands, COPY, DELETE, FDIR, PRINT, RENAME, VIEW, SETDEV and QUIT, the cursor being on FDIR.

Press on FDIR and the windows change, the main now contains file_names of the current disc/cartridge. The lower window blanks out, waiting for a command selection such as VIEW when it will show a file_name and in the main window, this same file_name, which is the last in the listing, is high-lighted. Pressing ENTER brings up a prompt in the lower window of 'y/n', selecting 'y' the main window fills with either data or a program listing. A byte count is given in the lower window, on the left. Selecting 'n' will abort the command, returning to the Command window. Only one file is accessed with commands PRINT, VIEW & RENAME but other commands deal in single or multiple selections by 'marking' selected files, or all, with the spacebar. SETDEV selects other Source and Target Devices. File_name groups from the Source Device can be selected as SUB DIRectories should the device contain more than 160 File_names.

When the main window displays the contents of a volume, having selected the name FList, it shows the file-names in four columns across the screen. This may then be printed out onto paper but when I tried this my printout was in three columns even though it printed in condensed format. Another little quirk is the fact that after a disc/cartridge has been put through the program, in every case an 'Flist' file_name is appended to the existing directory which is very disconcerting when not expected. But, having seen the print out of three columns I thought it very remiss not to have the byte-count attached to each file_name to save having to put it through the program repeatedly for just this purpose.

Pressing QUIT returns the program to the original Menu listing of:

(1) FTidy128
(2) Config128
(3) SuperBASIC
(4) APM Sweeper
(5) Backup

Selecting Config128 enables the default settings to be altered. It displays the Device Range, (Source and Target), Mdv1/2, Flp1/2 and Ram1/2 with a Working and Loading facility device and Printer settings and also has a Command Window below the Main Window with commands: EDIT, DFLT, LOAD, SAVE & QUIT. Pressing QUIT returns to the above menu.

Selecting (4) puts the program into game mode and the nice little game of 'Anti-Personal Mine' Sweeper comes on to the screen. My innocent understanding of Minesweeping is one of discovering at a safe distance, (and very carefully), just where a mine might be situated but this games assumes a lot of legwork to actually 'jump' on them to see whether they will explode or not. Very quaint! I didn't manage to get very high success rates in my scoring; correction, I haven't scored anything; I'm legless.

All in all this simple? The program provides a comprehensive file handling facility which deals competently with the need to 'tidy-up' discs and cartridges in order to use them in the most efficient and economic manner. The price would seem to be very reasonable and includes p.& p.

John Reeves

FTIDY
(128K) (MDV/FLP)
£5 inc P&P

QBITS
29 Silver Street
Willingham
CAMBS
CB4 5LF

HELPLINE

A good friend of mine is currently running a home for the elderly and mentally disabled, unfortunatly due to government legislation, places that teach such individuals (NACRO and the like) are now closing down, the only avenue open to them now, for a structured lifestyle that is, is the hope that private enterprise will intervene.

It is the intention of Stuart (my friend and colleague) and myself to start such a project, but rather than teach basket weaving and sewing and the like, I thought it may be an idea to try to teach them computing, or at least computer literacy, I may fail to get through to any of the residents, but at least I would have tried, and you never know, one of them may turn out to be the computer genius of the future, it could happen.

My computing experience is rather limited, although I grew up with Sir Clive, metaphorically speaking, so I am used to the computers he designed up to and including the QL, it is this type of computer that I am intending to use for the residents of the home, as I already possess some software that can be used on this type of machine. Now to the crunch ... What I would like to ask of all of you who have been patient enough to have read this article thus far, is to help provide the home and new school with such machines out of the kindness of your hearts, and I mean that with the upmost sincerity, not patronisingly, anything will be of use, broken, smashed, or in (hopefully) perfect working order, expansions, disc drives, monitors, printers, software, really anything at all, but please, all QL related (any drives except commodore will work with the QL). Anything that is sent to me, even down to just a single microdrive of games, will be acknowledged by myself by return of post, and I will endeavour to get the residents to write a letter of thanks (well sign one at least!) to all who are kind enough to contribute.

Anyone who thinks they may experience difficulty in getting items over to us please don't hesitate to drop me a line at the address below and I will try to resolve the problem myself.

The address for all contributions and correspondence is as follows:- Brian Richardson, c/o 21, Firefly Close, Roundshaw Estate, Wallington, Surrey. SM6 9HE.

For, and on behalf of, Harvest Care Homes, Haverfordwest.

PC KEYBOARD INTERFACE

Recently, I was wondering just how much improvement there would have been had the QL been supplied with a full-size working keyboard. I had seen several at computer fairs and been quite impressed with the difference in feel, even from the early replacement keyboard I had purchased for the ZX81 back in 1982 (made by DK'Tronics I believe). As my typing output has increased and in common with many programmers, I have developed my own inevitable touch typing style (four fingers and a thumb), I did wonder whether or not it would be worth my forking out £90 on a new keyboard for my QL.

Well, my current QL has recently celebrated its 5th Birthday (my first QL developed a fault after 6 months) and so I thought that a celebratory gift was in order. So I purchased... yes, you guessed it a printer!! What does this have to do with keyboards?? Well, I sort of blew the printer by not protecting it with a spike suppressor (users beware) and so one splurge of electricity soon blew the transformer (this has distinct echoes of when I had to purchase a new TV last year...). I was quoted £70 - £75 for the transformer alone, and so I decided to see what sort of deal I could get from EEC. A little bartering, an offer of a review or two, and I soon managed to obtain a new printer at a cut price, and had an offer of a keyboard and interface on loan for review purposes.

This was too good to be true, and so I snatched at the chance. Within four days of my order (including a weekend) the printer and two keyboards arrived at my door by courier. This was what I call excellent service....

There was a slight delay in getting the actual keyboard interface (it arrived three days later) whilst Bill obtained a fresh supply of the little wonders from the manufacturer, Jürgen Falkenberg in Germany. Still, within ten minutes of opening the package, a quick glance at the instructions and a slight expression of wonderment at the little black box, the keyboard was up and running!!

So, what do you need to do to install a new keyboard?

Well, first of all you must disconnect the QL from the power supply, and then open up the QL, by removing eight screws. Once it has been opened, you will need to locate the 8049 co-processor (a short black chip on the left of the two microdrives. This has to be teased out of its socket by using a chip-extraction tool, or a flat bladed screwdriver. This is quite a dangerous operation and great care must be taken so as not to bend any of the little legs on the chip (you will need this again later). Now what to do depends upon the version of the interface you have purchased. It is supplied in two forms:

1) There is a small interface board which has to be pushed into the 8049 socket (the 8049 chip is then pushed back into the top of the interface. The board does however mean that parts of the original QL keyboard have to be removed, thus making the old keyboard inoperable. It is a tight fit, but following the supplied fitting instructions carefully should ensure that the QL can be screwed back together. A short lead must then be fed out of the QL's housing somewhere (this may require a little filing of the base). This lead ends in a DIN socket in which to plug the keyboard's lead.

2) With the other version of the interface, a small black box and ribbon cable is supplied. You simply push the one end of the ribbon cable into the 8049 socket, push the 8049 chip into the top of this, and fold it over inside the QL, so that it now feeds out of the back of the QL, above the two joystick ports. There is no need here to remove any of the original QL keyboard, which means that this can still be used if you so wish (could be useful for two player games!). The QL must then be carefully screwed back together. There may be problems here if the QL's screws are tightened too much as this may damage the ribbon cable, although a user could of course always file down the small pieces of plastic in the top of the QL which you will find press down on the cable. One advantage of this method is that not only can the original keyboard still be used, but so can the microdrives!! - If users do find that the microdrives are not working, try loosening some of the other long screws that hold the QL case together, as I suffered from this problem until I realised that the QL casing was slightly bent from the different torque on the screws.

This ribbon cable is then fed into the small black box, which houses the interface itself. The box simply snaps together, with a lead leading out of the box to the DIN connector for the keyboard.

As to which version of the interface you purchase, this is very much dependent on how you intend to use the original QL. I daresay that it is not worth purchasing the extra black box and ribbon cable if you intend to put the QL into a box (if anyone knows how to do this quite simply, would they please let the rest of us know??)

Onto keyboards..............

Any standard keyboard can be attached to the QL using the given interface. There are two DIP switches on the interface itself which can be moved dependent upon the type of keyboard used, and it can take a little time to get the combination just right. I used the two keyboards supplied by EEC, both of which used the same DIP settings.

The two supplied keyboards come in a 102 key or a 84 key layout. The 84 key version takes up considerably less desk space, but does have a much less positive feel to the keys. It is very much a personal preference as to which keyboard you choose (unless you already have one which you intend to use), and so I will jut earmark the differences.

The interface seems to try to ensure that any keyboards plugged into it are as close to the original keyboard as possible (in key layout, not keybounce which seems non-existent with this interface!). This does mean that certain of the keys may not quite match up with what is written on the keys (especially the pound and @ signs). Any QL users will find that most of the key positions are familiar - indeed only the ALT, cursors, backslash and pound keys are in a different place.

The 102 key keyboard is in fact a Swedish keyboard with stick-on overlays to replace some of the Swedish symbols. I did find however that certain of these overlays did appear on the wrong keys, and so with a pair of scissors to hand this was soon sorted out. The glue on the overlays does not appear particularly strong, although I found that a dab of Pritt did the job. You hardly notice the overlays once they are in place (although the ones I cut to get the symbols in the correct places do have a distinctive feel to them!), and you are soon ready to begin typing ten to the dozen. The separate numeric keypad is a big help, and it is nice to see that using CAPS LOCK does not prevent you from using the normal set of numbers.

The keyboard is very robust, and quite a large size to accommodate all of the keys. As with the smaller keyboard, this has lights to indicate CAPSLOCK, NUMLOCK (toggles the Numeric Keypad between numbers and cursors), and SCROLL LOCK (the latter is for when the scroll lock key is pressed, which simulates CTRL F5). The extra keys on the larger keyboard are:-

a separate cursor key layout (enabling you to use both the Numeric Keypad and the cursors),

12 Function keys (rather than the usual 10 - the extra two do not have any effect on the QL),

a set of six buttons which emulate Toolkit II's ALT+ENTER, CTRL+RIGHT, ALT+LEFT and ALT+RIGHT (useful for Minerva users!!).

Both keyboards have single keys to emulate CTRL+LEFT, CTRL+F5, and CTRL+C. The extra keys (only five are not really supported) are very useful. However, the larger keyboard may infuriate games players a little, since on the smaller keyboard, some of the keys emulate diagonal cursor movements, which are missing on the 102 key job. Indeed, it seems a lot harder to move diagonally on the larger keyboard by pressing the normal down+left keys, than on the original QL (a joystick would be a big asset!!). The layout of the cursor keys will take some getting used to if you are a QL'er who is quite adept at using the four keys either side of the space bar, but then that is only a minor drawback (and a matter of practise).

I did (once or twice) experience a case where using Minerva's soft reset routine, locked out the new keyboard, but this was soon rectified by doing a full reset. Users may like to know however that the interface itself provides a soft reset (this is a full reset rather than Minerva's short cut), by pressing CTRL+ALT+DEL. This combination is unlikely to be used by any software and so should be quite a safe bet.

In both cases the keyboards are far superior to that supplied with the QL, and are certainly 500% better than a certain Philips keyboard I tried in a branch of the Midland Bank recently (it had a feel like a QL keyboard with baked beans under each key)!!!

This is a highly recommended upgrade to the QL, and should make it feel more like the powerful machine that it is.

<u>Rich Mellor</u>

Interface (Internal Fitting) £75
Case and Lead for External fitting £17
Keyboard 102 keys, UK version £25
Interface and Keyboard £90

Carriage £9, Postage £3, Overseas - Add £10 for delivery

(EEC) W.N.Richardson & Co.
18-21, Misbourne House
Chiltern Hill
Chalfont St.Peter
SL9 9UE
Tel:0753 888866 Fax:0753 887149

VISION MIXER PLUS

A year ago Dilwyn Jones marketed "VISION MIXER" written by himself and J.J. Haftke. It was reviewed in the Jan 91 issue of QL World. The reviewing author gave it a good no-nonsense pat on the back ("a very professional display ...... stand comparison with any commercial product") but there was little fire in its recommendation and I for one was not tempted to send a tenner in the direction of Bangor.

What I gleaned from that first review was that you could view screens in any sequence you wished and for as long as you wished - like a slide show - and between each screen you could do the fancy stuff in the form of DEFII and TOTP and RAPIDO etc on the box. You could have "wipes" and "zebras" and 'diagonals" and "stripes" and "spirals" and so on.

Once you decided up on a sequence of screen dumps (and chose the inter-screen Vision Mixer effect that would separate each of them) you could stand back to enjoy an endless graphic sideshow. Cor, let me at it!

This year DJC has pushed out a new version. VISION MIXER (VM) becomes VISIONMIXER PLUS (VM+) which does not supersede but rather complements the former. There IS one mighty big difference between the two. VM holds the graphic screens in computer memory : VM+ doesn't, but it accesses the screens directly from floppy or even hard disk. (A single 1440 sector disk will hold 22 screens.)

I had not used VM before, so my first meeting with VM+ was a new experience. I let the program do its own thing at first, displaying each of its sample screens. I noticed that the screens did not obscure channel Zero's space, leaving room for messages or notes or adverts. I later was to read in the clear manual that this is intentional to improve flexibility. By comparison, VM screens fill the whole available screen! Six of the sample screens that came with VM+ were pictorial while another four were adverts for other software goodies from the Jones stable. Well,why not?

As I got used to navigating the menu and sampling the different courses I became more aware of the options,and I learned how to compose a picture show of my own. I could determine how long any screen would be displayed. I could choose any one of FIFTY effects to separate these screens. Some of the effects were moving graphics in their own account, quite independent of the screens they separated. Other effects used the next graphic screen, giving a more interesting, transparency build-up, see-through effect. I could choose the transition effects appropriate to the screens of my picture show or I could let the program do it for me in random fashion. Once I had punched in this programme of events,the system was tolerant of later changes.

Dilwyn Jones is commendably honest in drawing our attention to an operating restriction. He reminds the user that the screens of VM+ are held on medium and not in QL memory. The program must therefore continuously access that medium. That's no big thing, as long as you are not mounting an endless-tape type display in your shop window when you might prematurely tire out the floppy or hard disk. VM is the utility to use in a continuous environment. VM+ is the one that a lecturer or teacher would use.

My lecturing days are over, but I found in VM+ a solution to a problem that's been with me for some time now. How could I leaf through my Clip Art screens? I have two of Keith Reader's Clip-Art floppies. One has sporting graphics and the other has 'whimsies', sort of funny caricatures you can slip into a letter heading or use to advantage in a bit of desktop publishing. There are some two dozen full screens on each disk, far too many to call up manually one at a time. Too much effort and laborious Lbyting. VM+ did the trick. It took a one-off 10 minutes session to construct a once-and-for-all VM+ programme of events. Now I can at will leaf through a virtual scrap-book with much less hassle and labour.

VISION MIXER PLUS is a handy, reliable, slide-show utility without the fireworks that its title might imply. If you want to emulate the fantastic dynamic graphic Catherine Wheels, Galactic Rhomboids, Stardust Snowstorms and pixel gymnastics of the TV giants, then you are also obliged to emulate their software spending power and pay lots more than Dilwyn Jones is asking.

Jimmie Robb

VISION MIXER PLUS FLP/384K £22.50

DILWYN JONES COMPUTING
41 Bro Emrys
Tal-y-Bont
Bangor
Gwynedd
LL57 3YT

REMIND-ME

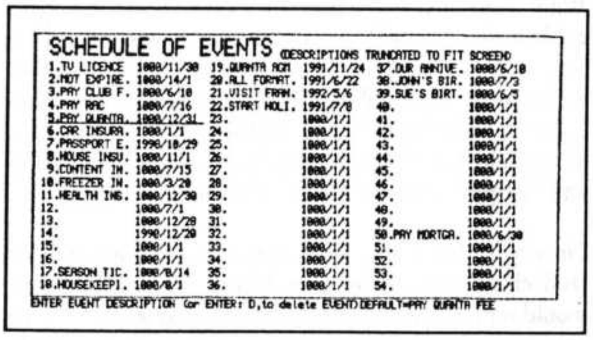
This nifty program, (a cartridge or disc), provides a fast, compact reminder of all those dates throughout the year whereon fall important events such as birthdays and anniversaries, or essential payments such as insurances.

It is composed of two main parts, the daily REPORT and the YEARLY SCHEDULE OF EVENTS. It has been compiled by way of Q_LIBERATOR, by Liberation Software which makes it 'zing' along, so much so that, although the program is intended to be 'multi-tasking', it finishes almost at once when run. It can be booted up alone with its own 'boot' program as given or executed with EXEC_W (drive)_REMIND_ME. Such is its speed it is the ideal medium for booting up a subsequent program following its use as a daily reminder of things to remember. The program needs no memory expansion and can be used on the standard 128K QL with micro-drives only. However, it needs a monitor to display, a TV screen is too small.

SCHEDULE: Up to 54 events can be catered for in the one year and all are numbered consecutively in the schedule with no regard to date order. This means that any unused number (or obsolescent), can be selected for any type of event which suggests that an initial organisation of type such as birthdays, anniversaries and periodic payments can be grouped on the schedule which is displayed on the one screen, the descriptions being a little truncated.

Events may be added to, deleted from and altered on the schedule using F5 to edit by selecting the relevant number using 0 to abort. This is always on offer following the REPORT, together with Space to end a run.

An interesting aspect of the REMIND-ME program is the way it deals with periodic monthly and annual events in that the number 1000 is used for yearly events and 0(zero) for the monthly. Having added the event to the schedule in this manner ensures that every month or each year the event is faithfully reported without further ado. To delete an event simply select the appropriate event number and enter a 'd' in the description column.



```
SCHEDULE OF EVENTS (DESCRIPTIONS TRUNCATED TO FIT SCREEN)
 1.TV LICENCE  1000/11/30  19.QUANTA AGM  1991/11/24  37.OUR ANNIVE. 1000/6/10
 2.MOT EXPIRE. 1000/14/1   20.ALL FORMAT. 1991/6/22   38.JOHN'S BIR. 1000/7/3
 3.PAY CLUB F. 1000/6/10   21.VISIT FRAN. 1992/5/6    39.SUE'S BIRT. 1000/6/5
 4.PAY RAC     1000/7/16   22.START HOLI. 1991/7/8    40.            1000/1/1
 5.PAY QUANTA. 1000/12/31  23.            1000/1/1    41.            1000/1/1
 6.CAR INSURA. 1000/1/1    24.            1000/1/1    42.            1000/1/1
 7.PASSPORT E. 1990/10/29  25.            1000/1/1    43.            1000/1/1
 8.HOUSE INSU. 1000/11/1   26.            1000/1/1    44.            1000/1/1
 9.CONTENT IN. 1000/7/15   27.            1000/1/1    45.            1000/1/1
10.FREEZER IN. 1000/3/20   28.            1000/1/1    46.            1000/1/1
11.HEALTH INS. 1000/12/30  29.            1000/1/1    47.            1000/1/1
12.            1000/7/1    30.            1000/1/1    48.            1000/1/1
13.            1000/12/28  31.            1000/1/1    49.            1000/1/1
14.            1990/12/20  32.            1000/1/1    50.PAY MORTGA. 1000/6/30
15.            1000/1/1    33.            1000/1/1    51.            1000/1/1
16.            1000/1/1    34.            1000/1/1    52.            1000/1/1
17.SEASON TIC. 1000/8/14   35.            1000/1/1    53.            1000/1/1
18.HOUSEKEEPI. 1000/8/1    36.            1000/1/1    54.            1000/1/1
ENTER EVENT DESCRIPTION (or ENTER: D,to delete EVENT)DEFAULT=PAY QUANTA FEE
```

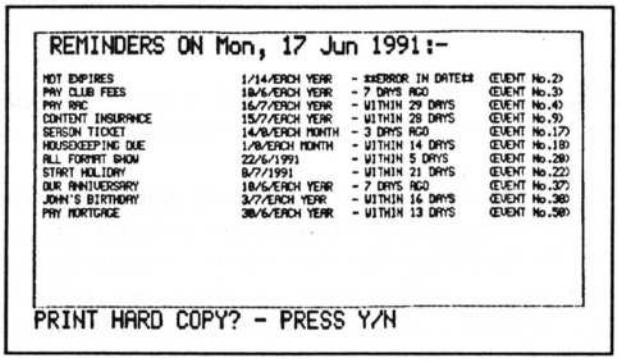
Having carefully set-up the SCHEDULE we can now utilise the:

REPORT: Given that there are EVENTS on the SCHEDULE the report is automatically produced in seconds, when running the program. Should there be no events a NIL report is produced.

The REPORT will list all EVENTS with due dates falling between seven(7)days before the current date and the end of the month following the current month. Thus, if the current date, (picked up automatically from battery-backed clocks or from the SDATE you have just inserted into the computer), falls on the first (1st) of the month, two complete calendar months of current and following events are reported plus seven days prior to the current date, ie., the last seven days of the previous month.

Unfortunately, as far as my pernickety mind is concerned the REPORT is produced in schedule number order and not date order which means that a little mental sorting out is required by the user, surely the raison d'etre for computers. This is balanced by each event reporting the number of days before being reached unless it has passed by.

A glaring omission I feel, is that there is no facility to append a <dealt with> symbol to a reported event that has been noted and attended to. Another raison d'etre?



```
REMINDERS ON Mon, 17 Jun 1991:-

MOT EXPIRES          1/14/EACH YEAR   - **ERROR IN DATE**  (EVENT No.2)
PAY CLUB FEES        10/6/EACH YEAR   - 7 DAYS AGO         (EVENT No.3)
PAY RAC              16/7/EACH YEAR   - WITHIN 29 DAYS     (EVENT No.4)
CONTENT INSURANCE    15/7/EACH YEAR   - WITHIN 28 DAYS     (EVENT No.9)
SEASON TICKET        14/8/EACH MONTH  - 3 DAYS AGO         (EVENT No.17)
HOUSEKEEPING DUE     1/8/EACH MONTH   - WITHIN 14 DAYS     (EVENT No.18)
ALL FORMAT SHOW      22/6/1991        - WITHIN 5 DAYS      (EVENT No.20)
START HOLIDAY        8/7/1991         - WITHIN 21 DAYS     (EVENT No.22)
OUR ANNIVERSARY      10/6/EACH YEAR   - 7 DAYS AGO         (EVENT No.37)
JOHN'S BIRTHDAY      3/7/EACH YEAR    - WITHIN 16 DAYS     (EVENT No.38)
PAY MORTGAGE         30/6/EACH YEAR   - WITHIN 13 DAYS     (EVENT No.50)

PRINT HARD COPY? - PRESS Y/N
```

GETTING STARTED: Make a working copy using the supplied R_clone_bas program, a simple task but be warned that you may copy only to MDV or FLP, not to both, on the same medium.

There is a pre-setup schedule for you to play with until you are satisfied you know how it all happens; you then delete the information, substituting your own.

There is a facility to printout hardcopy; the printer defaults together with other defaults such as Ser ports and baud rates are automatically configured(customised) during the first use of the program.

The manual advises that the best use of this program is as a preamble to other programs to ensure that you are always kept up to date with oncoming 'events' and the program gives a really fast 'boot-up' to subsequent programs; no time is squandered.

REMIND-ME-PLUS

This enhancement to REMIND-ME offers three improvements in addition to minor alterations and corrections.

1. The maximum number of events, which can be scheduled has been doubled from 54 to 108.

2. The maximum description length has been lengthened to 40 characters from 30.

3. The hardcopy option now contains a sub-option to printout the description in full 40 characters on a line-per-event layout. This ommits unscheduled and blank events to save time and paper.

A short program is supplied to transfer your existing SCHEDULE to the new program and the amendment note calls the program AMEND_bas but I think they mean ADAPT_bas.

THERE IS A GRAVE WARNING: "boot" and "MDVboot" are not the same as the REMIND-ME files - DO NOT CONFUSE THEM OR MIX.

John Reeves

CHANGING MODES

Many people have complained that they are unable to understand how the new MODE command on the Minerva ROM works and do not therefore get the most out of this feature.

You may already be aware that Minerva can operate in a dual screen mode whereby besides the normal QL screen which is normally used by programs, a second screen exists (at address $28000, where the system variables are normally stored). If you press F3 or F4 on start-up of Minerva (instead of the usual F1 or F2), Minerva is placed in dual screen mode.

You can toggle between the two screens at any time by pressing <CTRL><TAB> (although if Minerva is only in single screen mode, you will just see a lot of flashing colours - these are the system variables in action!).

The idea is that you can run, for example, two programs which use the whole screen at the same time, one on the normal QL screen (at $20000, known as scr0) and the other on the second screen (at $28000, known as scr1), without them corrupting each other. Okay, so you can already do this under the Pointer Interface, I hear you cry! Yes, but you try running a Mandelbrot program for instance, which can take an hour to draw a very complex screen and doing something else at the same time - as soon as you switch to a program which is trying to access a part of the screen used by the Mandelbrot program, the Pointer Interface makes the Mandelbrot program wait....

With Minerva's dual screen mode, you can happily run the Mandelbrot program on scr1 and something else, maybe a word-processor program on scr0 without stopping either one from working (subject to one or two limitations - see below).

Anyway, let me just recap first of all on how to use the original QL MODE command:

Normal QL ROMs only recognise two display modes, Low resolution (mode 8) and high resolution (mode 4). You can change between these two modes with the commands MODE 8 and MODE 4 respectively. Additional hardware (such as the THOR XVI computer) allows other modes, but that is beyond the scope of this article.

These original MODE commands still work on Minerva without any problems. I ought to point out that the MODE command will accept any integer in the range -32768..32767, but for the sake of compatibility with other operating systems, might I just point out that it would be sensible to stick to either MODE 8 or MODE 4 to choose the two main modes (I know of at least a MODE 2 and a MODE 12).

Normally, when the MODE command is issued, it alters everything running on the QL and forces a re-draw of all currently opened windows (and sometimes even some that aren't open on Minerva with Lightning, but then that's another story!!!). If Qjump's Window Manager is present, MODE will only affect the current Job.

There does also exist specialist software which allows a 'split-screen' whereby two modes can be displayed at the same time (the screen can only be split horizontally) - even this will work on Minerva without amendment.

Minerva extends this command so that it can be used to alter the various parameters for its dual screens, and even allows you to have a different mode on both screens. However, before I move onto this, a little word about compatibility.

Any Basic or well-written machine code programs should work fine in Minerva's dual screen mode. However, some programs and toolkit extensions assume that the screen is always located at address $20000 (131072) in memory (the screen can also have a different address on the THOR), in which case they will only be of any use to a program which is running on scr0. Some other programs (such as Quill and Turbocharged/ Supercharged programs) will not work in the dual screen mode at all and you will need to run Minerva in single screen mode for these programs. However, more and more programs are being altered to enable them to run in dual screen mode and the latest version of Qliberator has also been altered to ensure that Qliberated programs will work in this mode. Qview do supply a little program which can make a fair attempt at predicting whether or not a program will work in dual screen mode if you are interested.

One of the main drawbacks lies in an incompatibility problem with the Pointer Environment which will not allow you to have a different mode on the two screens and can make it very difficult for programs operating on scr1 (what is more, current versions do not recognise that the programs are running on separate screens and therefore still locks them out when you switch jobs). Hopefully Tony Tebby will be kind enough to update this in due course.

Lightning works fine in dual screen mode, although I believe that Speedscreen users may have to use the 'p' version.

In order to allow you to access all of these weird and wonderful options, Minerva accepts an enhanced form of the MODE command:

```
 MODE screen_mode [,display_type]
```

Although the display_type was actually intended for use on the original QL ROM, it was never actually supported in earlier versions of SuperBasic (although it could be used from machine code).

The display_type can be one of four values (the default is -1):

Display type	Effect
0	Set to monitor mode
1	Set to TV (625 lines) mode (European)
2	Set to TV (525 lines) mode (American)
-1	Leave display type as it is

Although the display_type may at first appear useful, changing it after the Pointer Environment has been loaded will create havoc (especially with any existing buttons!). However, in order to access the additional facilities afforded by Minerva, you will need to specify the display type (you therefore normally use a display type of -1).

If you do not specify a display type, the MODE command will only affect what is known as the 'Default Screen' (on start-up this is scr0). The second screen is known as the 'Other Screen'.

Now I get onto the fun bit.....

Minerva's extended MODE command allows you to specify the Default Screen. Each Job is given its own Default Screen (which is the current default when the Job was started), allowing each Job to alter its own Default Screen without affecting the rest of the system.

Now, just in case you were wondering, the Default Screen need not be the Displayed Screen (what you can see on your TV/Monitor when the QL is switched on).

This Default Screen is used whenever a new window is OPENed - subsequent accesses to that window will then only affect that screen. Other commands which access the Default Screen are MODE, VIEW, and COPY xxx TO scr_. All of the other commands which are used for graphics use a channel number and will therefore affect the screen where that channel is located (this is not necessarily so of some toolkit functions unfortunately, as some still expect the screen to be located at $20000 rather than using a machine code call to find out where the screen is which the channel is accessing). To toggle the Default Screen between scr0 and scr1, use MODE 64+32,-1.

I must just take this opportunity to warn you that a bug in Minerva can create havoc when opening new windows in dual screen mode. If the Default Screen is not the currently Displayed Screen, the newly opened window will have similar characteristics as if it had been opened in the Displayed Screen's mode - this is okay if both screens are in the same mode, but otherwise, you can end up with a lot of flashing text on the Default Screen. It is not possible (so far as I am aware) to find out which is the current Default Screen, or indeed which is the current Displayed Screen and in view of this bug, it is therefore important to keep a track of which is which (or write a little machine code function). Anyway, I use the public domain Fn toolkit in my programs (unfortunately I cannot find the document setting out the different commands and so cannot send this to be placed in the C.G.H. library..) which contains just such a function, DEFAULT_SCR which returns the number of the current DEFAULT_SCR (either 0 for scr0 or 1 for scr1). With this installed, you can now force the QL to display the Default Screen with the command:

```
MODE 16512+DEFAULT_SCR*16+(16128-256*16),-1
```

(I will explain these different parameters in due course).

Now onto the next important bit of information (I hope that you are managing to understand all of this!) - not only can a screen be a Default Screen or the Other Screen, the Displayed Screen or the not Displayed Screen, but it can also be Blank or Visible.

If a screen is Visible, it is just that. Assuming that both scr0 and scr1 are Visible, if you press <CTRL><TAB> you will switch between the two screens and be able to see what is happening (if anything) on them both. In contrast, if a screen is Blank, although anything can still be written to the screen, when you switch to that screen, you will not be able to see anything until it becomes Visible again. This allows background work to be carried out.

Some definitions might help:

Displayed Screen - this is the screen which is currently in front of the user on his/her monitor or TV.

Default Screen - the screen on which a program's windows will be opened and upon which the normal MODE 4 and MODE 8 commands will have an effect.

Other Screen - the opposite to the Default Screen (ie. if the Default Screen is scr0, then the Other Screen will be scr1).

Visible Screen - this means that the specified screen can actually be seen by the user.

Blank Screen - the specified screen is invisible to the user (allows background work to be carried out).

Whilst using Minerva's extended MODE command, it is worth bearing in mind that there is no forced re-draw of all the current windows unless you specify that this must be carried out (or if you use the original MODE variants).

Now, I will try to explain the new parameters introduced by Minerva for the display_mode. In order to do this, it is simpler to split it into two sections: toggling current values and setting absolute values.

Toggling the screen parameters

This uses the format: MODE 64+n,-1, where n has the following meaning:

n	Effect	From:	To
1	Toggle Other Screen :	Visible	Blank
2	Toggle Default Screen :	Visible	Blank
4	Toggle Other Screen :	4-colour	8-colour
8	Toggle Default Screen :	4-colour	8-colour
16	Toggle Displayed Screen:	scr0	scr1
32	Toggle Default Screen :	scr0	scr1

Adding together different values of n will combine these effects (although if one of the values is to be 32, the default screen will be toggled before anything else is carried out).

Some examples of its use are therefore:
MODE 64+16,-1:PAUSE MODE 64+16,-1 Show both screens.
MODE 64+4+8,-1 Toggle the mode of both screens

Setting absolute screen parameters

This can be extremely complex, and it is therefore highly important that you understand everything explained above before proceeding. This is made worse by the Minerva manual containing the incorrect formula, but I shall continue regardless..

To set the screen parameters ABSOLUTELY, you need to use the following form of the MODE command: MODE -16384+128+m+(16128-256*t)+c,-1.

Now, I would suggest that you leave the command in this format in your programs for later debugging, and so do not try to calculate the single number represented by the above complex formula.

In this formula, m, t, and c can have the following values:

t= t1 + t2 + t3 + ...
where, t1, t2, t3 etc. have the same values as n above. DO NOT however combine the values of n at this stage.

m=k1*t1 + k2*t2 + k3*t3 +.....
where k1, k2, k3 etc. have the following effect upon the corresponding values of t1,t2,t3, etc.:

k	Sets n to:
0	the 'from..' column above
1	the 'to..' column above
257	toggle value (as above)

c can have the following values, and is used to tell the MODE command whether or not to re-draw all currently opened windows on specified screens:

c	Effect
0	Do not redraw any screens
-16384	Re-draw the Other Screen
32768	Re-draw the Default Screen
16384	Re-draw both screens

Again, different effects can now be combined with relevant values for each t and k. If one of the values includes changing the Default Screen, again this will take precedence to all of the other commands.

An example to help you on your way:

To Blank the Other Screen and then force it into 4-colour mode and redraw all of the windows. First of all decide which attributes you want to alter:

t1=1 (change Other Screen from Visible to Blank or vice versa)
t2=4 (change Other Screen from mode 4 to mode 8 or vice versa)

t therefore = 1+4 (or 5)

Now you have to calculate m which shows whether you want to force the attribute to an absolute value:

1*t1 (make Other Screen Blank (use the to column))
0*t2 (make Other Screen mode 4 (use the from column))

m therefore = 1*1 + 0*4 (or 1)

Now to say whether the windows are to be re-drawn,

c=-16384 (redraw the Other Screen)

Hence, the command needed is:

```
MODE -16384+128+1+(16128-256*5)-16384,-1
```

As you can no doubt see, this dual screen MODE command is not the easiest of things to use, but once mastered, you will be able to extend the range of uses to which you put your QL and no doubt you will increase the amount of multitasking that you do.

Rich Mellor

DBEASY

For your money, you get a disk and an A4 sheet of paper on which is what passes as a summary of the program/s you have bought and a 'start-up' instruction. The disk label says it is DBEASY V1.5, a data-base made easy.

'DBEasy is a complete database system written in the ARCHIVE's programming language with easy-to-use menus for readily accessing most of ARCHIVE's features. No knowledge of ARCHIVE is needed to use it. DBEasy is a general purpose program that can be used for keeping track of anything. See the review of DBEasy by Bryan Davies in the May '91 issue of QL WORLD.' The afore-going is an extract from a file on the disk called 'INITIAL BOOT FOR CHAOS DISK', (something you might miss), and I am writing this to confirm that what is claimed is valid.

However, not so fast please! On the disk is a lot more than DBEasy and the A4 page(itself a README_doc on the disk) says it includes as a bonus CHAOS BUSTERS, an integrated system of software - a disk\micro directory program to list/print/search directories (DIRLUX_bas), an automatic database maker of disk/micro directories (FILEX_bas), and the friendly "Front End" database program (DBEasy itself) for ARCHIVE to let you access the directory database made by Filex. DBEasy has more general applications. It will let you create and use databases for any purpose you might have from simple menus. Most of ARCHIVE's powerful features are immediately available to you painlessly. But this isn't all, not by a long-shot. Also included on the disk is an ARITHMETIC trainer program, a TASKET program for independent multi-tasking within this disk, and a HELPER program, a set of useful toolkit-like functions in a menu format. Most deal with file handling.

Now, Mr Cable states that his self-proposed brief was to produce an easy entry into database building and operations for those duffers of us, myself included, who just couldn't get to grips with ARCHIVE in it's pristine form. I am happy and glad to say that he has done exactly that with DBEasy except for one thing; the hardcopy of 'how-to' instructions is a higgledy-piggledy nightmare of 'half-hidden' clues as to what step one should take next.

There are, on the disk, eight _doc files, (one of which is the README file which you have printed), that should be printed-out in order to make your life easier. There is also information contained in the file BOOT_INITIAL_bas which would be helpful when printed onto paper and more in the DBEasy program itself in files softfile_dbf and dbehelp_dbf in addition to snippets dotted here and there. But despite earnest searching I did not find 'how-to' information regarding the menu called up when using the - switch mode - function in a database. So where did I find this information? In the Bryan Davies review of version 1.2, (this version is 1.5), published in the May 1991 issue of the QL World magazine. Do get hold of a copy.

It really will pay dividends to go slowly on this one and get your act together before you frustrate yourself. In the first place DBEasy can only be run inside of the ARCHIVE program, copy it onto your backup disk. (The INITIAL_BOOT will assist you in this). Next, you must decide whether you want to initialise TK2_ext in which case copy this or have it available. Then you can add ALTkey presses for QUILL so copy Quill and if you're copying that you might as well copy the other two, Abacus and Easel. (But be warned; if you now contemplate copying Exchange a minor problem exists in that you have to slightly modify the program DBEmm_pro, the manual shows how). If you have Lightening then copy that also. Then of course, you must have a printer_dat, copy it. By now you've got a pretty full disk which means that after tutoring yourself with the help of the seven demo databases, you might consider pruning them on your working-disk to give yourself some elbow room.

Start by either a reset which auto-boots or Lrun the Boot. A menu of 1 to 8 choices asks where the CHAOS_BUSTER disk is, (Mdv,Flp,Ram), select a number and another menu asks where you want a scratch pad, (Ram1_ is the obvious choice). Pressing a key gives a little wire-animation to be followed by a date/time check. The initialising process above is now arranged and finally you arrive at the CHAOS BUSTER's main menu 1 to 0 (ten options), DIRLUX, FILEX, EXEC ARCHIVE, Tasket, HELPER, run a boot program, Display instructions, Print instructions, Restart CHAOS, Exit CHAOS.

The obvious is to start DBEasy which means you Exec ARCHIVE, wait for it to come on screen, type RUN, press <ENTER> then type into the two sets of inverted commas, <""> DBEasy and press <ENTER> again. You are straight away presented with another menu and careful study will show you that you are already inside a pre-determined database. The program has selected the last database used, having recorded it, and awaits your choice of option to either use this database or switch to another. The menu headings are: Single mode - Multiple mode - Backup - Directory - Switch - Program Device - Screen Colors. Choosing the Switch mode reveals a list of all the current databases currently held on available disk in the log. To select one of these files the Use option in the sub-menu is chosen by an arrow and you are asked to choose an option by a number, or a letter and when you do, nothing seems to happen except that a highlight bar has now moved from the database it was on to your chosen one. You must now 'exit' from where you were and by so doing you gain entry into your selected base. Weird, definitely weird.

The ten text, six numeric and two fixed field record is accessed for browsing by the arrows, first, end, next, previous or by number of record or by 'string'. You can add, delete, alter records with ease. Add or clone a blank database to the log. You can configure printout inside the program and customize your screen. Once you have mastered the intricacies of the program the hardest work is to construct your database by filling in the fields of your records but, once this has been done, what joy in being able to manipulate your chosen information in exactly the way you want it!

The author, Bill Cable, didn't mention it but the most important thing this program does is to give confidence. It shows you that ARCHIVE is within your reach and if you put in a bit more effort, with DBEasy's help, you can master the whole thing. To this end I recommend that you read, in Sep 1992 QL World magazine, an article starting on page 37, 'One Man's System' by David McCullagh, (in particular the last few paragraphs on page 39), '...and I find it surprising that no one has made a serious attempt at producing a better Archive/DBase-type of database for the QL.' But it really is academic now unless you're curious; in DBEasy the hard work has been done and for peanuts, (note the rarefied prices McCullagh is quoting as against the fifteen pounds asked for DBEasy).

John Reeves

DBEasy
FLP/512K
£15

Dilwyn Jones Computing
41 Bro Emrys
Tal-y-Bont
Bangor
Gwynedd
LL57 3YT

SCRIPTWRITER

Andy Pritchard's SCRIPTWRITER is a program which can help you write the cinematic masterpiece you have dreamed of! Basically a specialised word-processing program, it automates the nitty-gritty tasks of typing in characters' names in full, adjusting spacing and indentation, signalling stage directions, and numbering pages and scenes. It does not provide you with a plot; nor will it tell you whether your characters' lines are actually sayable in front of an audience!

(However, consult the file STYLE_pwd, and there you will find quite a lot of very useful advice!)

Although specialised, SCRIPTWRITER claims to double as a plain ordinary word-processor suitable for writing plays, books, or stories- in fact, this review was written with it. In this role, SCRIPTWRITER does most of the basic things you expect of a word- processor, but inevitably misses out some of the more advanced facilities that you get with Text87 or Perfection. There are some fairly basic facilities that you don't get, too: you can't right-justify on screen, for instance (though you can right justify printout via printer control characters), and you can't increase the number of characters you get per line on screen. Nor can you reformat a paragraph like this one. In fact, writing text other than TV/film scripts would be much more easily accomplished using an ordinary word-processor; SCRIPTWRITER is best kept for the specialised task of preparing scripts.

The program is SuperBasic compiled with Digital Precision's Turbo, and works quite fast enough for my typing fingers. (Unlike Quill, SCRIPTWRITER does not slow down with large files.) It is a development of an earlier program, QL Playwright, which received a good review in QL World (October 89). There are many improvements, the most significant of which is probably the incorporation of the print routine within the main program.

FIRST IMPRESSIONS

The playing area is divided into three and a quarter windows. At the top of the screen is a black rectangle, giving space for eighteen lines of 72 characters' width, where you write your script. This is framed at the sides by borders of parallel vertical lines - rather like curtains drawn aside from a proscenium, in fact! - and below by a Menu window (on the left), and the general-purpose command-and-response window (on the right); these are both 5 lines deep. The Menu window shows the menus in general, or in particular - the Main menu details the other five menus, headed Script menu, Modify menu, Search menu, Header details, and File menu, which you can cycle through to see what's what. The other window gives further information about the actions you want, or the results of what you have already done. Oh, yes, the quarter window: the date and time are shown towards the top of the curtain stage left.

All this is very nicely presented in green and black, with white text on a black background. The command windows are green on black, with black on white for highlighting commands, and white on red used for warnings. Care has evidently gone into the cosmetic aspects of the program. On starting, the program searches for a character set which, if found, will replace the standard QL set. It's sturdier and clearer, and good to look upon. You can replace the file with a set of your own devising, if you wish. You also get on-screen italics - useful for stage directions, for instance. Again the program searches for this file on starting, and again you can replace it.

SPECIFIC APPLICATIONS

The primary reason for buying this program, of course, is to write scripts with it. (If you want ordinary word-processing you will be better off, if you can afford them, with Text87 or Perfection.) The program scores against its possible competitors heavily by simplifying the repetitive special tasks associated with script writing. The key to these functions is held in the "Header details" menu. This menu allows you to specify the Title, the Roles, the Indentation, and Scene text for you script. Thus:

> - Opt for "Title", and you are prompted for the Title of your script, obviously, but also for your name and address - all the information, in fact, which will appear neatly tabulated on the front cover of your script. (I would prefer the cursor to appear at the beginnings of lines here, though, to indicate that you are expected to delete or over-write the prompts "Title", "Address1" etc.)
>
> - Opt for "Roles", and you are prompted for the names of characters, and you are prompted to enter the names and roles of characters, separated by a hyphen, as in "Pritchard-Scriptwriter".
>
> - Opt for "Indentation", and you are invited to change the default settings for indentation of scenes, dialogue, etc. It's probably better to leave them as they are until you're really sure how you want them.

(The layout above is what you get by setting blocks to "Dialogue" indentation, which is more or less instantaneous with blocks of such size.)

SCRIPTWRITER in Practice

For a practical test, I typed in a bit of an actual script - not mine, in fact, since I lack the necessary abilities, but a snatch of a famous if slightly old-fashioned play, coyly referred to by actors as "the Scottish play" because of the superstition that the most incredible bad luck follows from mentioning its real title. Naturally, wild horses wouldn't drag the title out of me here. This is how one famous scene looks under SCRIPTWRITER:

Enter Macbeth

```
                  LADY MACBETH
My husband!
                  MACBETH
I have done the deed. Didst thou not hear a noise?
                  LADY MACBETH
I heard the owl scream and the crickets
cry.
                  MACBETH
When?
                  LADY MACBETH
Now.
                  MACBETH
As I descended?
                  LADY MACBETH
Ay.
                  MACBETH
Hark!
                  LADY MACBETH
Who lies i'th second chamber?
                  MACBETH
Donalbain.
                  LADY MACBETH
This is a sorry sight.
        [Looking at his hands.]
                  LADY MACBETH
A foolish thought, to say a sorry sight.
```

All this is quite automatic. You press F4 for dialogue, F5 for the s/d "Looking on his hands' (I also italicised that with the CTRL/I toggle), and F2 for the Action "Enter Macbeth", and the cursor immediately moves to a sensible place. (Since the program makes specific provision for different tab settings, TAB is reserved for the Hard-Space, shown as a tall rectangle.) The Character control, F3 is more clever. That calls up a list of characters, and if you type "ma" or "lad" the full name is printed in the right place. Since the program did not offer me Macduff instead of Macbeth, it must take the names in order, and not check for confusions.

Notice that the script above is not set out as you would normally expect to find it in a book, i.e. with character names abbreviated at the left. That would mean mixing different indentation settings on a single line - something I have not been able to do and I suspect that the program will not allow it. That is, you can't put a dialogue-indent on the same line as a character-indent, for instance. Unsurprisingly, SCRIPTWRITER cannot make allowance for the fact that the Scottish play is in verse! Andy Pritchard's choice of layout is determined by the specific format of TV and film scripts.

BUGS AND PROBLEMS

There are one or two bugs, I think, and several problems. There are, for a start, a number of things SCRIPTWRITER won't do. These are the facilities that I felt I missed

No insert/over-write toggle (an intentional omission - but not therefore an advantage!);
No facility to extend lines beyond 72 characters;
No facility to tidy up untidy paragraphs in text without any special indent set.

If such things matter to you, it may deter you from buying the program. But, of course, not everyone needs these things: certainly you don't need more than 72 characters per line to write a script.

The program assumes that one of the indentation settings will be used at all times except when importing text such as SuperBasic programs. If you do not use the indentations, you can run into more problems than losing the CTRL/F command: when you type onto the end of a paragraph of plain text followed by another, as I have done here, the text runs into the beginning of the next paragraph. The moral is: always use the Function Key indentations.

Now to the bugs:

a) SCRIPTWRITER wouldn't run on my usual messy system, with QPACII etc etc., but crashed with a very pretty pattern every time. This may perhaps be an effect of a SuperBasic POKE which I haven't had the energy to explore the causes of failure of. So I ran the program on a "clean" machine instead, and it worked beautifully. SCRIPTWRITER clearly does multitask, however, since it is designed to do so alongside its printout task. If you suffer from the POKE-crash, you can prevent it by deleting the Italic _font file from the disk - though this means that you won't get italics on screen, and you will also lose the pretty red arrow which shows marked lines.

b) I had trouble with deleting a line with the delete key(s). When you delete back to the end of a preceding line, the cursor can finish up in some strange places on the page. NB The SCRIPTWRITER Delete Line command (CTRL/D) operates perfectly nor mally, in contrast.

c) When I first tried the program on my "cleaned" machine, I got a crash message, saying "W2_TXT: old version". I do not know why, since the version I have seems to be 2.2, and the message, if I understand the SuperBasic program correctly, relates to versions earlier than 1.5. Oddly, when I had backed up the program, the message no longer appeared. The original disk still refuses to execute the program: the copy consistently works! Odd.

d) The Directory command failed to recognise any of my devices as valid. Flp1_, Ram1_ and Win1_ were all snubbed with the message "Invalid device". Why? The program loads/saves normally.

e) Having got quite a long way with this document, and saved it, I was unable to load it back again, because of an error ("out of range" at Line 20730). After trying this and that I finally had recourse to the trusty Editor! ... Back in SCRIPTWRITER, I entered a few brief test files, and was able to save and reload them successfully. I have no idea why the review file failed to do so. Therefore... I cleaned up the control characters with Editor, saved the review, and imported it successfully back into SCRIPTWRITER. It was very, very easy to reformat it to its original form. This impressed me, though it shouldn't have been necessary!

COMMANDS & FACILITIES

The range of facilities is impressive, though not exhaustive. There are comprehensive navigation key-presses - though ALT/LEFT and ALT/RIGHT do not have what I have come to regard as the almost standard effect of moving the cursor to the beginning or the end of the line; instead, they circle round the menus. SHIFT/arrows instead have the "Home-End" effect.

All the commands are available immediately as either CTRL/letter commands, or Function Key commands. Although the program shows menus, from which you can access the highlighted commands by pressing SHIFT/ENTER, I found it easier to use the menus and highlights as reminders of which keys are used for immediate commands. Some commands do not appear on a menu at all, so you do need to read the manual, which is disk-based, brief and clear, with no more than a sprinkling of mis-prints, or call up the Help screen to jog your memory. The command-letters chosen are "natural", and therefore easy to remember. The Function keys give you different indentations, as indicated previously. Shifted, they allow you to indent marked blocks of text. There is also an immediate command CTRL/Z , which removes any indentation inserted in error.

The CTRL/letter commands include:

Goto bottom/top/marker/next scene/last scene
Italics on/off
Format a paragraph
Change case of word
Cut and paste
Mark position (a nice little red arrow appears)
Display Help screen
File/Printout commands
Delete line
Search and Replace (individual or global)
Refresh screen
Reset indentations to default values
Add Header details (characters etc.)
Quit

These all seem to work as intended. I particularly like the speed and simplicity of the way you mark and operate on blocks of text. You can't set a mark in the middle of a line, but it is not a facility that is often required anyway. The two italicizing commands also impress me; they make it very simple to insert italic print, or to italicize existing text, anywhere in the document.

PRINT ROUTINE

The print routine is configurable, but supplied as Epson-compatible. If your printer is Epson-compatible, italics on screen will come out as italics on paper.

The print routine is intelligent, and avoids widow and orphan lines at page-breaks. That is, it avoids the idiocy of, say, one word of a speech at the end or beginning of a page. It also does handy things like present your script in a style acceptable to prospective publishers. Your title is nicely printed at the start, with your name and address, and a list of the characters in your masterpiece. Page numbering starts automatically when the script or other work starts. Believe it or not, I have not tested the program on a full-length script. It seemed to operate well on the short test-pieces I used.

It is very easy to configure the print routine to your requirements by means of a "Parameters" option. The program gives you avery nice feeling of being in control of things.

There are several useful facilities built into the program. You can, for example, print a range of files (acts or scenes, say). You can screen a list of files and choose which ones to print at print time. You can opt for the background print routine which will multitask and allow you to continue working on the final act of your script while the first two acts are being typed. You are invited to choose whether to send the output to screen, printer, or file. All very handy!

MEMORY REQUIREMENTS

Obviously, the amount of memory you need is dependent on the length of the script you want to write. Digital Precision supply a DATASPACE task which adjusts what it says it adjusts, so you can tailor SCRIPTWRITER to the memory available to you. The program itself occupies 46k, and as supplied assumes a 512k memory expansion. If the dataspace is reduced for smaller memory, your script will need to be saved as a range of separate files. On a basic machine, you may have to think in terms of very short scenes, too! SCRIPTWRITER is therefore not recommended for use on an unexpanded QL.

CONCLUSIONS

This is in general a very handy and very user-friendly program. It does most of what it says it does, most of the time. However, reliability is, I think, the over-riding requirement of a utility program of this kind. I suspect that the bugs can be ironed out without too much difficulty. They certainly need to be. SCRIPTWRITER could do with being more compatible. My system is messy, but by no means unique, and I'm unlikely to use a program much unless it fits in with the system.

Otherwise, the program appears extremely well thought out, with every effort paid to making things simple for the user. It is easy to get used to the key presses and the logic of the program. This is essential, to leave the grey cells free to pursue the flight of fancy and the depth of the human heart. Anyone who can use Quill can learn SCRIPTWRITER very quickly, and stay in a much better temper, too.

The program continues to be developed. Version 2.3 has removed some bugs in version 2.2 - I shall look forward to version 2.4!

Mike Edwards

SCRIPTWRITER £12.50
FLP/MDV/128K+

C.G.H. Services
Cwm Gwen Hall
Pencader,Dyfed
Cymru
SA39 9HA

SUPER TOOLKIT II - PT IV
(REVISITED)

9 Job Control

The QL was the first affordable micro to allow multitasking. It is one of the many features that sets it apart from the herd and yet is one of the most difficult to control satisfactorily on a standard machine.

TKII provides four commands (JOBS, RJOB, SPJOB and AJOB) and four functions (PJOB, OJOB, JOB$ and NXJOB) to control jobs.

9.1 Job Control Commands

JOBS lists the current jobs. By default the output will go to #1 but as with standard SuperBASIC procedures the output may be sent to any other channel by simply appending channel number in the normal manner. Thus JOBS #2 will display a list of jobs in #2.

TKII also allows the use of **implicit channels**. That is, to send the output to a device it is not necessary to open a channel to that device, send the output to the channel (as shown above) then close the channel, the command may be appended with \ and the device name. The following will create a file job_txt on flp1_ that contains the list of jobs on the system:

```
JOBS \flp1_job_txt
```

Assuming a printer is attached to ser1, it is just as easy to print out the list of jobs with one command:

```
JOBS \ser1
```

With SuperBASIC as the only job on the machine, the JOBS command would display a table as follows:

```
Job tag   owner  priority
0   0     0      32
```

SuperBASIC does not have a job name. Names are normally displayed after the job priority. If a job is suspended (QDOS will suspend a job when resources it wishes to use are unavailable to it) then an 's' would be shown immediately to the left of the job priority.

For an explanation of job number, tag, identification (id), owner and priority see section at end.

Note QDOS accepts any negative value of job id to signify the current job. This is true of the following job control procedures and functions. It is accepted, however, that -1 should be used to signify the id of the current job: consistency greatly improves the clarity of programs.

RJOB is used to remove a job (other than SuperBASIC) from the machine. RJOB is followed either by the job name or by the job id. The job id may be quoted as **job number, job tag** or as a single number equal to 65536 * tag + job number. Job number and tag are displayed using the JOBS command. If a program, mandelbrot, is executed using the command ex flp1_mandelbrot, the JOBS command would produce a list as follows:

```
Job tag   owner  priority
0   0     0      32
1   0     0      8   mandelbrot
```

Note, a job activated by SuperBASIC will start with a priority of 8. However, many jobs will change their own priority after activation. Some jobs will not have names.

The job mandelbrot may be removed using the command **RJOB mandelbrot**.

There is a further parameter that may be added to the command. This is an error code which through the use of machine code could be read by the parent job. This is not relevant to the user of SuperBASIC except that it appears that when RJOB is passed the job id (or job number and job tag) rather than the job name, the error code must also be used. This is understandable: if two numeric parameters are supplied QDOS would not always be able to decide if the parameters are the job id and error code or the job number and tag. Therefore, two parameter are considered to be job id and error code, while three parameters are taken to be job number, job tag and error code. The reason for using the job id as a parameter is that as previously mentioned, some jobs do not have names. Another reason is that it is possible to have may jobs with the same name. If the job name is used then QDOS will remove the first of that name.

SPJOB allows a job's priority to be altered. As already mentioned, a job started by SuperBASIC will be given a priority of 8. If the job is required to run faster or slower then one must raise or lower its priority. Like RJOB the first parameter is either the job name or job id. If a job id of -1 is used then the current jobs priority is altered. (If one is typing SPJOB as a direct command this would be SuperBASIC. However, if SPJOB is used in a SuperBASIC program that is then compiled an argument of -1 would refer to this compiled program). The second parameter is the priority. This is an integer between 0 and 127. A priority of 0 means the job will become inactive (it will not get a share of cpu time).

Thus, if one wished to alter the priority of the job mandelbrot (as used in the example above) one could type either of the following:

```
SPJOB mandelbrot,16
SPJOB 1, 0, 16
```

The job would then be expected to run faster provided other jobs were active on the machine. If no other jobs are active then it makes no difference if the priority is set to 1 or 127. How much faster it runs depends upon the total priority of all other jobs on the machine and the availability of resources that it wishes to use. For example only one job may use the keyboard at a given time so if a job requires input and the keyboard is already used then it will be suspended by QDOS until the keyboard is available (either because the first job has finished or control-C is used).

AJOB is used to activate a program which has been loaded into memory but not previously started. If a job has previously had its priority set to 0 it could be reactivated either by setting its priority to a positive value or by using AJOB. A job executed with the command ET would be in an inactive state until activated by further commands such as AJOB.

9.2 Job Status Functions

PJOB returns the priority of a job. As with the commands above, the job may be specified either by its name or by its id. Since it is a function the job name or id must be enclosed in brackets. Thus one may type **PRINT PJOB (mandelbrot)**. From within the program, mandelbrot, one might wish to double the job priority:

```
150 PRINT 'Do you wish to speed up mandelbrot?'
160 answer$ = INKEY$ (-1)
170 IF answer = = 'Y' THEN
180   priority = PJOB (-1)
190   priority = 2 * priority
200   SPJOB -1, priority
210 END IF
```

OJOB returns the id of a jobs owner (ie the job from which it is activated if EW or EXEC_W was used, or SuperBASIC if EX or EXEC was used). In the above example, **PRINT OJOB (mandelbrot)** would print 0, the id of SuperBASIC.

JOB$** returns the name of a job given its id. Thus, **PRINT JOB$ (1, 0) would print 'mandelbrot'. This is useful in any programs that refer to other jobs: one may wish to use job ids in calculations but when it comes to displaying information it is better to convert to the job name.

NXJOB returns the id of the next job in the job tree. Using the same example as above, **NXJOB (0)** would have a value of 1 ie the next job after SuperBASIC is mandelbrot.

Super Toolkit II thus provides a set of commands and functions for controlling jobs and finding out information about jobs. It would be nice to have more functions: for example a function that returned the location of a job in memory, its length, the location and length of data it is using etc. Much of this information can be found by using machine code.

MULTITASKING UNDER QDOS

Job ids, numbers and tags

Each job that is run on the QL is given a unique job identification which may be quoted as a combination of a job number and job tag or as a single number which is equal to the job tag * 65536 + job number. SuperBASIC is job 0 (ie job number and tag and therefore job id set to 0). The first job after SuperBASIC is given a job number of 1 and job tag of 0. Successive jobs are given an incremented job tag and a job number set to the lowest unused job number. Suppose the first two jobs are started on the machine and then the list of jobs is displayed:

```
EX mandelbrot
EX prime
JOBS
```

the result would be:

```
Job tag    owner  priority
0   0      0      32
1   0      0      8  mandelbrot
2   1      0      8  prime
```

If mandelbrot is now removed:

```
RJOB mandelbrot
JOBS
```

the display would be:

```
Job tag    owner  priority
0   0      0      32
2   1      0      8  prime
```

Once a job's id has been set it does not change. If a new program, trig, is executed the job display would be as follows:

```
Job tag    owner  priority
0   0      0      32
1   2      0      8  trig
2   1      0      8  prime
```

The new job assumes the lowest available job number (the one vacated by mandelbrot) while its tag is one higher than for the previous job. The next job would have both a tag and job number of three.

Job Ownership

QDOS allows jobs to be executed in two ways. Using EXEC or EX the new job runs independently of the job from which it was executed. SuperBASIC is then the parent (owner) of the new job. If execution is performed using EXEC_W or EW then the new job does not run independently of the job from which it was executed. In this case, the job that performed the EXEC_W or EW command becomes the parent (owner) of the new job and will be suspended until the new job (child) has terminated. Whereas the function OJOB will return the job id of a job's owner the JOB command will show the job number of the owning job. However, this is enough to uniquely identify the owning job since it must be one currently on the machine: if a job is removed its child jobs are terminated too. It is possible using machine code to execute a job such that it has a parent other than SuperBASIC (in manner of EW) and that the parent is not suspended (in manner of EX) while the child is running.

When a series of filters is executed using the EX command (as explained in part 3 section 8.2) eg **EX add3_bin, in_dat TO mul2_bin, out_dat** the jobs will run simultaneously and add3 will be the parent of mul2. So the chain of jobs may be removed by removing the first one in the chain.

Job Priority

A part of QDOS called the scheduler determines at any given instant which job is to be run by the cpu. The scheduler is activated by a hardware interrupt occuring every 20ms (it is linked to 50Hz mains cycle). When a job is executed a **job control block** is set up immediately before the program code in memory. The job control block is 102 bytes long and contains information about the job. Two bytes in this block contain information about job priority: one byte labelled **PRINC** contains the value of the jobs priority as set by SPJOB or as given to the job on execution; the other **PRIOR** is the current value of the accumulted priority. When a new job is executed PRINC is set to 8 (but may be modified once the job is running) whereas PRIOR will initially be 0. Each time the scheduler is invoked it increments each jobs accumulated priority (PRIOR) by the value of PRINC. Then the job with the highest value of PRIOR is executed for one time slice (the remainder of the 20ms until the next invocation of the scheduler) and has PRIOR set to zero. This system allows all jobs capable of running to gain a share of cpu time in roughly in proportion to their priorities (PRINC).

With just two jobs running it appears the ratio of the two job priorities has to be greater than n:1 for the ratio of share of cpu time to be n:1 eg if the ratio of priorities is 1.9:1 then the ratio of cpu times will be 1:1, if the ratio of priorities is 3.1:1 the ratio of cpu times will be 3:1.

When the ratio of priorities are in a ratio of 2:1 then two different situations could occur. When the values of PRIOR are the same for each job (as will happen at set intervals when the ratio of priorities is a whole number) if the job with the highest value of PRINC was to run then the ratio of speeds would be 2:1. If, however the job with the lower value of PRINC was to run then the ratio of speeds would be 1:1. Experimentation has shown the latter to be the case and the following example illustrates what is happening. T is the time slice number (number of 50ms intervals that have passed). PRIOR A and PRIOR B are the accumulated priorities for job A and job B respectively. N CPU is the number of times that the job has been active on the CPU.

	JOB A (PRINC = 63)		JOB B (PRINC = 126)		
T	PRIOR B	N CPU	PRIOR B	N CPU	
0	0	0	0	0	
1	63	0	126	0	
1	63	0	0	1	JOB B runs, PRIOR B reset
2	126	0	126	1	
2	0	1	126	1	JOB A runs, PRIOR A reset
3	63	1	252	1	
3	63	1	0	2	JOB B runs, PRIOR B reset
4	126	1	126	2	
4	0	2	126	2	JOB A runs, PRIOR A reset
5	63	2	252	2	
5	63	2	0	3	JOB B runs, PRIOR B reset
6	126	2	126	3	
6	0	3	126	3	JOB A runs, PRIOR A reset

After six time slices each job has run three times. The values of PRIOR for the two jobs every other time slice. Each jobs run on altenate time slices.

Now suppose the priorities are in a ratio greater than 2:1. If JOB A has priority 63 and JOB B has priority 127 then the ratio of job priorities is 2.016:1 not a great deal different from the above?

JOB A (PRINC = 63) JOB B (PRINC = 127)

T	PRIOR A	N CPU	PRIOR B	N CPU	
0	0	0	0	0	
1	63	0	127	0	
1	63	0	0	1	JOB B runs, PRIOR B reset
2	126	0	127	1	
2	126	0	0	2	JOB B runs, PRIOR B reset
3	189	1	127	2	
3	0	1	127	2	JOB A runs, PRIOR A reset
4	63	1	254	2	
4	63	1	0	3	JOB B runs, PRIOR B reset
5	126	1	127	3	
5	126	1	0	4	JOB B runs, PRIOR B reset
6	189	1	127	4	
6	0	2	127	4	JOB A runs, PRIOR A reset

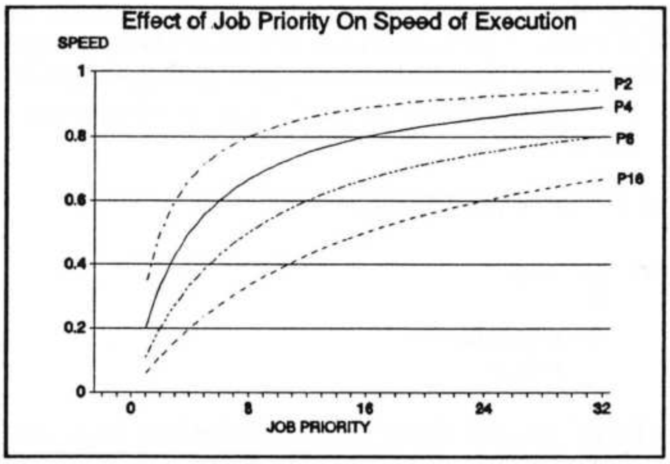
After six time slices JOB A has run twice and JOB B has run four times. he pattern now repeats. The ratio of time on the cpu is now 2:1!

So, with only two jobs on the machine it is just as well to set one at priority **one**. Then if the second job is required to run **n** times as fast (that is get **n** times as much cpu time) its priority would be set to **n + 1**. If the first job is set to priority **p** then the second may be set to any value between **n * p + 1** and **(n + 1) * p**. eg if p is 12 then the second job's priority may be set to any value between 37 (= 3 * 12 + 1) and 48 (= (3 + 1) * 12) for the ratio of job speeds to be three. A short SuperBASIC function could be used to give the ratio of job speed given the job priorities:

```
1000 DEFine FuNction RSpeed (PriA, PriB)
1010 :
1020  RPri = PriA / PriB
1030  IF RPri < 1 : RPri = 1 / RPri
1040  IF RPri = INT (RPri)
1050    RETurn RPri - 1
1060  ELSE
1070    RETurn INT (RPRI)
1080  END IF
1090 END DEFine RSpeed
```

With more jobs running the situation is more complex and I have not investigated it fully. However, it is a fair approximation that a job gets a share of cpu time in proportion to its priority. Suppose there are three jobs running on the system, BASIC at priority 32, MANDELBROT at priority 8 and MATHS priority 16. MATHS could be expected to get roughly $^{16}/_{56}$ (0.29) of the cpu time. If the job MATHS has its priority doubled it would get roughly $^{32}/_{72}$ (0.44). Thus, a doubling of job priority does not yield a doubling of job speed.

The graph on the next page shows how a job's speed varies with its priority. The four lines indicate four different loads on the machine. For example, P2 indicates how a job's speed will vary as its priority is altered when all other jobs on the machine have a priority totalling two.



Suppose the job being considered is BASIC. The speed of BASIC is measured by timing how long a given program takes to run. Suppose that there are three jobs active on the computer:

Job	tag	owner	priority	
0	0	0	32	
1	0	0	1	mandelbrot
2	1	0	1	prime

The curve P2 shows how the speed of BASIC would be expected to vary as its priority is changed. If the BASIC program takes one minute to run when no other jobs are active on the machine then with the jobs shown in the table above the program might be expected to take nearer 64 seconds (ie 60 / 0.94). As should be expected it is possible to run low priority jobs without having much affect on the main jobs (in this case BASIC). Suppose under these circumstances the job mandelbrot takes two hours to run and the job prime also takes two hours and that the BASIC program would be run several times over so as to be running all of the time that the other jobs are running. With BASIC at priority 32 it gets about 94% of the cpu time and the other two jobs get about 3% each. Suppose the priority of BASIC is reduced to 8. The graph (still line P2 - the priority of mandelbrot and prime together) indicates that BASIC would then get about 80% of the cpu time and so the same BASIC program would take about 75 seconds to run. Now mandelbrot and prime would each have about 10% of the cpu time and so would be expected to run more than three times faster than they did previously: they would each take about 36 minutes to run rather than two hours!

Well that's the theory. To test it I ran a BASIC program which calculates 10000 sines, prints how long it took and then repeats. This takes about 63 seconds with no other jobs running. The I ran a similar program (only 1000 sines) under a MultiBASIC (this is a facility provided under Minerva which allows additional BASIC interpreters to be invoked) at the same time. The table below shows the timings for the SuperBASIC program as its priority is varied and the MultiBASIC's priority is kept at 2. The theoretical times taken from curve P2 are shown in parenthesis.

Priority	SuperBASIC		MultiBASIC	
32	68	(67)	102	(105)
16	73	(71)	57	(57)
8	85	(79)	25	(25)
4	127	(95)	13	(19)
1	127	(189)	13	(9)

The theoretical values are calculated by dividing the how long the job takes when no other jobs are running by the speed index read from the graph eg the SuperBASIC program takes 63 seconds with no other jobs running. With MultiBASIC running at priority 2 and SuperBASIC running at priority 32 a speed of 0.94 is indicated (Taking a vertical through job priority 32 to the curve P2. Where they intercept take a horizontal to the y-axis and read off the speed value). This value may also be calculated by dividing BASIC's priority (32) by the total priority of all the jobs (32+2) ie 32/34=0.94. The SuperBASIC program takes 63 seconds to run when it is the only job on the machine. So, when it has 0.94 of the cpu time it will take 63/0.94=67 seconds. If SuperBASIC has 0.94 of cpu time then MultiBASIC will have 0.06 of cpu time. Its program takes 6.3 seconds to run with no other jobs active on the machine. So, under these conditions it will take 6.3/.06=105 seconds.

Pretty good agreement! Except for the last two. As explained previously when the ratio of job priorities is between 1:1 and 1:2 inclusive the ratio of share of cpu times will be 1:1. Both when the SuperBASIC priority is 4 with the MultiBASIC priority at 2 and when the SuperBASIC priority is 1 with MultiBASIC priority at 2, the ratio of priorities is 2:1 (or 1:2 it doesn't matter) and so they get equal shares of cpu time and each job takes twice as long as if they were the only job on the machine.

Suppose that the priority of all the jobs is now changed:

Job	tag	owner	priority	
0	0	0	1	
1	0	0	8	mandelbrot
2	1	0	8	prime

Now the way the speed of the BASIC program changes as the priority is varied, is shown by line P16 since the other jobs on the machine have a total priority of 16. With BASICS priority set to one it will get around 6% of cpu time. If its priority is increased to two it will get about 11% of cpu time and if it is increased to 4 it would get about 20% of cpu time. So at low priorities the speed of the BASIC program will be roughly proportional to priority.

So the important points are:

1 Only when a job's priority is low compared with the total priority of all other jobs on the machine will a change in priority yield a (roughly) proportional change in speed.

2 When a job's priority is high compared with the total priority of all other jobs on the machine then a change in priority will have relatively little affect upon the jobs speed but will affect the speed of other jobs significantly.

Jobs should only be assigned priorities between 0 (inactive) and 127. It is possible to assign priorities up to 255 but this is not specified in the QDOS documentation and is not very worthwhile for the reasons just explained.

COMPARISON WITH OTHER OPERATING SYSTEMS

In this part comparison is only made with VMS since the other operating system I have previously shown, MS-DOS, is not intrinsically multitasking. I have had no experience of using any of the multitasking front ends such as Microsoft Windows or Quarterdeck's DESQview.

	QDOS	VMS
List jobs (1)	JOBS	SHOW SYSTEM
Remove job (2)	RJOB 1,0,-1	STOP PROCESS/IDENTIFICATION=2D0
Set priority	SPJOB 1,0,16	SET PROCESS/IDENTIFICATION=2D0/PRIORITY=3
Activate job	AJOB 1,0	SET PROCESS/IDENTIFICATION=2D0/START
Read job priority	priority = PJOB (1,0)	priority = F$GETJPI ("000002D0","PRIB")
Read job name (3)	name$ = JOB$ (1,0)	name = F$GETJPI ("000002D0","PRCNAM")

1 VMS accepts abbreviations for standard commands eg SHOW SYS and SHOW PROC/ID=2D0

2 Under VMS each new process is given a unique process identification number which starts at 1 and is incremented for each execution. It is quoted in hexadecimal.

3 The VMS 'lexical' function F$GETJPI (get job process information) can return many different pieces of information about a process depending upon the arguments passed to it. Information about a job may also be displayed using **SHOW PROC/ID=20.**

Stephen Bedford

SUPER TOOLKIT II - PT V

SuperBASIC PROGRAMMING

In the first part of this series it was noted that the facilities provided by TKII are divided in to two main sections those that extend SuperBASIC as a command language and those that extend it as a programming language. The commands covered in this part of the notes, and in those parts that follow, would most often be used within a SuperBASIC program. Many of the extensions offer new ways of doing things that may achieved using the standard SuperBASIC commands. These new commands provide substantial improvements and it is worth while getting in to the habit of using them. Other commands provide features that are not available under standard SuperBASIC. So TKII increases both the flexibility and power of SuperBASIC as a programming language.

10 Open and Close

10.1 Open Commands

The standard commands OPEN, OPEN_IN and OPEN_NEW have been modified so as to use the data default directory. In addition OPEN_NEW has been modified so as to prompt if the file already exists. Suppose the disk in flp1_ contains the following files and the data default is set to flp1_:

```
TKII1_doc
TKII2_doc
TKII3_doc
```

Then typing

```
OPEN_NEW #3, TKII3_doc
```

would result in the following prompt

```
FLP1_TKII3_doc exists, OK to overwrite..Y or N?
```

The new command **OPEN_OVER** will overwrite an existing file without prompting but otherwise works the same way as OPEN_NEW.

The command **OPEN_DIR** appears to have no use on a normal QL. No error message is given if the name passed does not match an existing file name, or part file name, and yet a file is opened for read only by this command. With a Gold Card real sub-directories are implemented. Using OPEN_DIR it is possible to view a directory file, however this is no different from using OPEN_IN. I expect that the command is designed for use with the QL Hard Disk or with some planned (at the time TKII was written) upgrade to QDOS.

Note the various forms of the OPEN command may also be used to open a channel to a device such as a serial port. In this case the commands all function as though OPEN had been used (ie OPEN, OPEN_IN, OPEN_NEW and OPEN_OVER will all perform the same function). Therefore, only OPEN should be used for such devices. The type of device will determine whether read or write access is allowed not the form of the OPEN command used. eg

```
OPEN #3, ser1
```

will allow read and write access to the serial port via channel #3 whereas

```
OPEN #4, scr
```

will allow only write access to the screen window #4. The channel should be opened to con rather than scr to allow read as well as write access. Thus, it should be apparent that it is only sensible to use the OPEN command (rather than any of its variants) when dealing non-directory devices. (Directory devices are those which support files eg microdrives, floppy disks, hard disks etc).

10.2 File Status

A new function, **FTEST**, returns a code indicating the status of a file or device:

0	**file exists**
-3	**out of memory**
-6	**no room in channel table**
-7	**not found**
-9	**in use**
-12	**bad name**

Note in the TKII manual it states that -11 is returned for bad name. However, the error codes listed under error handling on page 19 of the Concepts section of the QL User Guide show 12 as bad name and 11 as drive full. (See example v below).

The following examples demonstrate the conditions under which the various codes will be produced. For all examples assume that the same disk as above is used, that no other programs are accessing the files and that the data default is still set to flp1_.

```
i) PRINT FTEST (tkii2_doc)
```

This would result in 0 being printed: the file exists.

```
ii) OPEN_IN #3, tkii_doc
    PRINT FTEST (tkii2_doc)
```

Again 0 would be printed. The function FTEST has succeeded in checking the existence of a file by opening it for read access then closing: more than one channel may be open to a file provided they only allow read access.

```
iii) OPEN #3, tkii_doc
     PRINT FTEST (tkii_doc)
```

With the file already open for both read and write access, FTEST will return the code **-9**: the file exists but is in use. No other channel may be opened to the file either for read or write access while **#3** remains open. Under theses conditions WSTAT would give the following:

```
tkii1_doc
    16470   1991 Feb 08 17:54:41
tkii2_doc
    in use
tkii3_doc
    15330   1991 Feb 08 18:14:39
```

WSTAT is unable to read the file details from its header.

```
iv) PRINT FTEST (tkii3)
```

This would result in -7 being printed to indicate that the file cannot be found: the extension _doc has been left off.

```
v) PRINT FTEST ('file name is longer than 40 characters')
```

Would result in **-12** being printed since the file name is greater than forty characters long (once the data default 'flp1_' has been added to the start!). Spaces are allowed in the file name provided it is enclosed in quotes.

The number of channels that may be open on the QL is dependent upon available memory. There seems to be an upper limit (over 300!) to the number of channels that may be opened even if there is plenty of memory available. I can open channels 3 to 347 inclusive either to **scr** or to a file (using OPEN_IN to give shared access) while Quill is running. Having all these channels open uses about an extra 25K of memory. Of course 347 may the limit to the number of channels that may be open to SuperBASIC and not to QDOS. The point is that if FTEST returns code

-6 (no room in channel table) then something strange is going on!

I have never seen the error code -3 (out of memory) returned by FTEST - I used to keep an eye on the amount of free memory available when I had an unexpanded machine and now I have plenty of memory it is not a problem.

10.3 File Open Functions

TKII also provides a number of functions for opening files that correspond to the commands described in 10.2 above. The functions return an error code indicating either successful opening of the file or the reason for failure. Optionally, the functions may be used to find the lowest unused channel. The functions are **FOPEN, FOP_IN, FOP_NEW, FOP_OVER** and **FOP_DIR**. The use of each function should be apparent from the description of the corresponding commands in the previous sections. FOPEN is explained further.

FOPEN may be passed either one or two parameters. When two parameters are passed these are taken to be the the channel number and the file (or device) name. eg

```
error = FOPEN (#3, flp1_boot)
```

The value of **error** will be 0 if this operation succeeds or a positive integer indicating the error code should it fail. The circumstances under which errors will be generated are similar as for using FTEST (see section 10.2). In the examples that follow assume that the data default is set to flp1_ and that it contains the files:

TKII1_doc
TKII2_doc
TKII3_doc

```
 i) error = FOPEN (#3, TKII1_doc)
```

error will be set to 0. TKII1_doc exists and has been successfully opened both for reading and writing.

```
ii) error1 = FOP_IN (#3, TKII2_doc)
    error2 = FOP_IN (#4, TKII2_doc)
```

Both error1 and error2 will be set to 0 since both channels are open to the same file for read access only.

```
iii) error1 = FOPEN (#3, TKII3_doc)
     error2 = FOPEN (#4, TKII3_doc)
```

In this case, error1 will be set to 0 but error2 will be set to -9 (in use). It is not possible to open multiple channels to a file when one of those channels allows both read and write access.

```
 iv) error1 = FOPEN (#3, TKII4_doc)
```

error will be set to -7 (not found) since the file does not exist on flp1_.

Page 19 (error handling) of the Concepts section of the QL User Guide lists all of the standard errors. Alternatively, the following will display the list of errors on the screen:

```
FOR N = -1 TO -22 STEP -1 : PRINT N, : REPORT N : PAUSE 50
```

The **REPORT** command is part of TKII and will be covered in section 17, although its function will be apparent from entering the above line.

The following is an example of a situation when FOPEN might return a non-zero error code:

```
error1 = FOPEN (#3, flp1_boot)
error2 = FOPEN (#4, flp1_boot)
```

Assuming that no other channels are open to flp1_boot then error1 will be set to zero indicating that channel #3 has been successfully opened to the file. However error2 will be set to -9 (in use) since it is not possible to have two or more channels open to a file when one of those channels allows write access to the file. If in both of the above commands FOPEN was replaced with FOP_IN then both error1 and error2 would be set to zero - both channels have only read access to the file.

FOPEN may also be passed a single parameter that indicates the file name. In this case the function returns a negative integer as above if an error occurs but returns a positive integer if the operation succeeded. This positive integer is the channel number. This will be the lowest unused channel number. Assuming that only the standard channels #0, #1 and #2 are open then

```
        chan = FOPEN (flp1_boot)
```

would open flp1_boot for input and output and chan would be the channel number assigned by the operation and would be set to 3. Assuming of course, that the file exists (for a new file FOP_NEW should be used).

The advantages of the OPEN functions over the commands are most apparent when being used in programs. The most obvious advantage is that they allow the program to trap the errors that would otherwise stop program execution. The programmer can trap errors that are beyond his control. For example, if a program requires to OPEN a file contained on a disk in flp1_ it may be that the user forgets to put the disk in the drive or puts in the incorrect disk. In this case the OPEN function will return -7 meaning **not found.** If such an error occurs the program can inform the user and prompt for the correct disk to be entered.

The second advantage of the OPEN functions is that there is no need to keep track of actual channel numbers but just variables representing the channel numbers. If a set of often used PROCedures and FuNctions are kept in separate files and MERGEd in to programs that require them there will be no need to check for possible conflicting channels if the OPEN functions are used with just one parameter. Also, by returning the **lowest** unused channel number the use of these functions ensures that the least amount of memory is used. This is because the channel table, (the area of memory that holds information about all SuperBASIC channels), uses an amount of memory determined not by the number of channels in use but by the value of the highest channel number. If channel #100 is used in a program then space is allocated for channels #3 to #99 even if they are not used.

10.4 CLOSE

TKII explains the additional features of the CLOSE command. I would like to add a word of warning about using CLOSE with no parameters appended. I often use this as a direct command when I am testing something however, I would not use it in a program. The reasons being that it does not add to the clarity of the program: by appending channel numbers to CLOSE it is made obvious which channels were open. This can be a help both to someone other than the original writer who is trying to understand the program and also to the writer in order to be sure in his or her mind exactly what operations the CLOSE is performing.

11 File Information

A number of functions are provided which return information about files. All of these functions accept either channel numbers (eg #3) or implicit channels (eg \flp1_boot) as a parameter.

FLEN returns the length of a file in bytes. This will be the same as shown by the WSTAT command. Thus, to find the length of flp1_boot, assuming the data default directory is set to flp1_, either of the following could be used:

```
        PRINT FLEN (\boot)
```

or

```
        chan = FOP_IN (boot)
        PRINT FLEN (#chan)
        CLOSE #chan
```

Note that FLEN requires only to read the file header so write access is denied by using FOP_IN. This reduces the possibility of error should CLOSE #3 be omitted by mistake.

Of course, the result may also be assigned to a variable:

```
boot_length = FLEN (\boot)
```

(See section 5.2 of Part 2 of these notes for further information about file lengths).

The function **FTYP** returns the file type. In the TKII manual it states that the function returns 0 for ordinary files, 1 for executable programs and 2 for relocatable machine code. However it appears that 1 is returned for executable programs and 0 for all other file types (except real sub-directories on Gold Card, QL Hard Disk etc).

Thus, provided that convention is being followed

```
PRINT FTYP (\boot)
```

would display 0. Files that contain QDOS or SuperBASIC extensions and are often given a name ending in **_ext** or **_rext** (indicating **ext**ension or **r**esident **ext**ension) are also of type 0. These are files containing relocatable machine code.

Executable programs (ones executed using EX and EW and equivalents) are of type 1. Thus it would be expected that the following command would display 1

```
PRINT FTYP (\quill)
```

The real sub-directories on the Gold Card have a file type of 255.

The function **FDAT** will return the amount of data space that an executable program uses once it has been activated. For other files (ie type 0) FDAT will return 0.

The TKII manual states that **FXTRA** finds the file extra info. What does this mean? I don't know. For most files this function returns 0 and for some it returns a large number. (On my working copies the same large number is returned for quill, abacus and easel, 1.329725E9, whereas the function returns 0 for archive. I don't know if it is relevant but the number returned for quill etc corresponds to a date of 2003 Feb 20 ... - perhaps FXTRA is reading a part of the file header which was planned to hold the creation or backup date but was not implemented)
The function **FNAME$** returns the name of the file. The following

```
chan = FOP_IN (flp1_boot)
PRINT FNAME$ (#chan)
CLOSE #chan
```

would result in **boot** being displayed. The full filename excluding the device name is returned.

Supposing a disk in flp1_ contained the following files:

```
letters_RICHARDALEXANDER_T87
letters_MIRACLESYSTEMS_T87
letters_TONYTEBBY_T87
addresses_RICHARDALEXANDER_T87
addresses_MIRACLESYSTEMS_T87
addresses_TONYTEBBY_T87
```

If the data defaults directory is set to flp1_letters (data_use flp1_letters or ddown letters) it would be possible to refer to letters_TONYTEBBY_T87 as TONYTEBBY_T87. The full filename may be found as follows:

```
PRINT FNAME$ (\TONYTEBBY_T87)
```

This would result in **letters_TONYTEBBY_T87** being displayed. The function could also be used by opening an explicit channel as in the previous example.

The last function in this section, **FUPDT**, returns the update date of the file (ie the date that file was last written to).

The update date is the only date maintained in the standard QDOS file headers (a backup date is also maintained in the headers of files contained on the QL Hard Disk and files under the Atari QL emulator). The function returns the date in the normal floating point format (see DATE in Keywords section of QL User Guide). To view this as a string containing year, month etc the result should be passed to the function DATE$. For example

PRINT DATE$ (FUPDT (\boot))

might display **1991 Jun 09 21:01:41**. Note the update date is not even set if TKII is not used, it will always be set to zero - 1961 Jan 01 00:00:00.

COMPARISON WITH OTHER OPERATING SYSTEMS AND PROGRAMMING LANGUAGES

Now that the notes are covering extensions to SuperBASIC as a programming language it will often be more appropriate to compare commands with similar ones available in other programming languages rather than operating system command languages. However, some of the features offered by TKII would be achieved by other programming languages by calling the operating system from within the program. Alternatively they may be provided by some non-standard feature - in programming languages other than BASIC the most commands and functions are covered by well established standards.

	SuperBASIC	**Pascal**	**C**
Terminology	channel	file pointer	file pointer

Open a file
(1,2,3)
	OPEN #3, flp1_boot
	OPEN_IN
	OPEN_OVER
	chan = FOPEN (boot)
	chan = FOP_IN (boot)
	chan = FOP_OVER (boot)
		update (fp)
		reset (fp)
		rewrite (fp)
			fp = fopen ("flp1_boot", "r+")
			fp = fopen ("flp1_boot", "r")
			fp = fopen ("flp1_boot", "w+")

Close a file	CLOSE #3	close (fp)	close (fp)
File Status	status = FTEST (\boot)	**(note 2)**	**(note 4)**

(1) Each of these Pascal statements must be preceeded by the statement which associates the file pointer, fp, with the file - **assign (fp, "flp1_boot").**

(2) ISO standard Pascal is rather short of error handling features. However, good Pascal compilers (such as Prospero's Pro Pascal which is available on the QL) have extra functions defined for checking file status etc.

(3) The C fopen functions return the file pointer or NULL (zero) if an error occurs.

(4) I have not delved into C enough (yet) to know if it has an equivalent to this function.

	QDOS	**VMS**

File info
(1,2)
	len = FLEN (\boot)
	date = FUPDT (\boot)
		len = F$FILE_ATTRIBUTES ("LOGIN.COM", "EOF")
		date = F$FILE_ATTRIBUTES ("LOGIN.COM", "RDT")

(1) Other operating systems, such as Unix, can probably give information like this about files but I am not familiar enough with them to know the appropriate function names. Most operating systems will give (optionally) details such as file lengths, dates etc in directory listings.

(2) This VMS function can accept many other parameters other than those shown here (EOF - gives block number of end of file; RDT - gives revision date and time). As with all VMS commands, abbreviations may be used.

Stephen Bedford

PICTUREMASTER PLUS

When I purchased my QL I had a small photographic business, and my use of the computer was purely for accountancy and word-processing with the aid of Abacus, Archive and Quill. It had occurred to me that a monitor might be quite a good way of advertising, – but I was no artist, and consequently steered clear of all QL graphics programs which were then available.

It was not until much later when I saw "Vision Mixer" being used by a trader to advertise his wares at a Quanta workshop, that I sat up and took notice! The screen effects, I was given to understand, were prepared from a program called "Picturemaster", and as all the basic hard work had already been done by Joe Haftke, even someone like myself would be capable of producing good results from it.

"Picturemaster Plus" is an upgrade of this program, and incorporates Joe's "Pad and Pen" concept for screen painting. This is analogous to painting a wall using a decorator's pad as opposed to a brush or roller. Using different shaped faces such as box, triangle, ellipse or circle, infinite design variations may be drawn and trailed - either filled or showing only the outer rims.

The resulting images can be condensed, magnified, repeated and moved. Any paint colour from the QL range can be chosen from an on-screen chart and work can always be "undone" without affecting previous work when a mistake is made.

Text, including shadowed lettering up to eight times CSIZE 2 can be used and three auto-centralised caption lines are available. "Handwriting" using a small box can be accomplished, and a grid of horizontal lines called-up to keep writing a straight line.

Screens can be stored in memory and recalled for blending with others. 45 predesigned screens are now available and a continuous display of them can be generated.

I have had a lot of fun using this program. Just as I originally bought the QL purely for business purposes and subsequently became "hooked" on computing for pleasure, – likewise with this excellent program from Dilwyn Jones Computing !

Jim Buik

PICTUREMASTER PLUS
FLP/384K
£20.00

Dilwyn Jones Computing
41 Bro Emrys
Tal-y-Bont
Bangor
Gwynedd
LL57 3YT

TEL: 0248 354023

PD NEWS

Is it really a year since I did the last of these columns ? Seems like only yesterday. (No, not really!) Anyway as I'm not paid by the word I'd better get a move on with updating you all on what's new on the P.D. scene.

The C68 for QDOS system is undergoing continual improvement, and we're currently shipping V3.05 with 3 Runtime and 2 Source Code disks (these latter being in ARC format.) These are backed up by a C Tutorial disk, Libcurses (1.01) by Keith Walker, Libcport (1.02) - a C-PORT Support Library disk, and QPTR for C68. There are also a couple more disks, including a C Programmers Disk, a C Debugging Disk and CFIX for CPORT - but we haven't received these yet but will make them available a.s.a.p.

One area where we get quite a few progs are the Device Utilities Disks. These have now spread onto two disks, the second one featuring ARC, Arce, Compact/Uncompact and Zip/Unzip all of which are file compression progs so you can squeeze more onto your disks. There's also several front-ends, fast file copiers and hard-disk utility progs on these two disks.

Those of you into Editing (and writing) programs may be interested to know that Richard Kettlewell has ported over version 3.11 of MICROEMACS which we've put onto Editors Disks 3 and 4. Jan Bredenbeek's excellent QED resides on Editors Disk 2 together with the runtime version of MICROEMACS (V3.10) - the latest version takes up the whole disk.

If you need to keep track of your disk collection then you may welcome Alan Pemberton's DiskTidy program, which will gather all the filenames of each disk - allowing you to find them later. Now if only programmers would use the program's name as a filename! There's a version of the C.G.H. Services catalog on this disk to show what can be done. DiskTidy is SQLUGWare - meaning that if you register with the author you'll get upgrades and a manual, any profits going to the Scottish QL User Group.

Those of you with good memories may remember me bemoaning the fact that the QL lacked a Spectrum emulator, especially as we could port graphics over and other 68000 machines had such emulators. Well, they're here! Carlo Delhez has produced both a Spectrum and ZX81 emulator and Davide Santchiara has done a Spectrum emulator. All on one disk - you lucky people! (Actually there's a later version of ZM1+ on Emulators disk 2). All progs are Shareware which means that if you cough up the readies you'll be kept up-to-date with changes and, if you send enough, you get the commercial versions of the progs which are faster and/or more compatible. Thanks to Simon N. Goodwin we've ported over 2 disks of "P.D." for the ZX81 from an Atari ST Library with file names in a format that Xtricator can read.

Thanks to Sohail Bhatty and Steve Johnson we can now offer a very wide range of P.D. fonts for Text 87, including Times, LGothic, Helvet, Cinema, Blippo and similar. These are in addition to Andy Dean's fonts. (If you have Font Disk 1 - send it back for an upgrade.) There are numerous disks of fonts in a variety of formats for the ST (and other machines) perhaps someone who understands these could do an article on how to convert them into a form that QL programs can use.

Fractals continue to amuse many people and we have added a brilliant disk by Per-Erik Forssen of a Lyapunov Space fractal generator together with numerous sample screens. Have to admit that this one is slow - anyone fancy doing a pure machine code generator ? Come to think of it many of our fractal programs are hamstrung by being written in compiled SuperBasic - the core of these programs isn't huge - any chance some m/code programmers lending the fractal people a hand ? (Be interesting to see what Miracle System's new graphics card can handle - high resolution and plenty of colours - just right for fractals!) (Personally I'll wait for a Falcon 030 at £375 from C.G.H. Services when Atari get round to releasing the 1MB version.)

The SuperBasic etc disk now incorporates Machine Code utilities as we've added Philip Holiday's 68000 DisAssembler and Jan Bredenbeek's QL Multimon (V2.1). Also on this disk is Laurence Reeves' QView Tiny Toolkit which, I suspect, will mainly be of use to those of you with a Minerva ROM. I've also added Laurence's FORTH to Programming Disk 9. (Thanks to David Gilham and Tony Firshman for these.)

Special mention must be made to Stan Harle who has bombarded me with a stream (sorry about the mixed metaphor!) of P.D. progs over the past year (not complaining Stan!) including quite a few Ham Radio and Money progs. Also added to the Ham Radio Progs disk is a Packet Radio program by P.M. Hounslow and M.J. Tribe. This needs a TNC2 type terminal mode controller and will only work with a TNC with RXBLOCK command. (Should that be Terminal Node Controller?)

Right that's a brief rundown of what's new at C.G.H. Services Public Domain and Shareware Library. If you do use any Shareware progs PLEASE send the requested donation or registration fee to the authors - otherwise we may not see such programs in the future. We welcome all additions to the library, even if it does sometimes take a wee while for me to check them out. If you are not intending to support your code, please send the source code for your programs so that other people can have a go at keeping them abreast of changes in QL operating systems. Apart from that - keep 'em coming, the quality of QL P.D. software is improving (no doubt due, in part, to the impossibility of sustaining low-level commercial publishing on the machine) and the price is a bargain: £2.00 per disk inclusive of media, postage and packing and a printed catalogue. Please add 10% for orders in Europe, 20% for the rest of the world. (Note - price of upgrading existing disks is £1.00 per disk if you send the original disk and it still works!)

Cheers

Richard Alexander